4 Classical Statistical Inference

"Frequentist or Bayesian? Flip a coin. (And observe the long-term outcome.)" (Frequentists)

This chapter introduces the main concepts of *statistical inference*, or drawing conclusions from data. There are three main types of inference:

- **Point estimation**: What is the best estimate for a model parameter θ, based on the available data?
- **Confidence estimation**: How confident should we be in our point estimate?
- **Hypothesis testing**: Are data at hand consistent with a given hypothesis or model?

There are two major statistical paradigms which address the statistical inference questions: the classical, or *frequentist* paradigm, and the *Bayesian* paradigm (despite the often-used adjective "classical," historically the frequentist paradigm was developed after the Bayesian paradigm). While most of statistics and machine learning is based on the classical paradigm, Bayesian techniques are being embraced by the statistical and scientific communities at an ever-increasing pace. These two paradigms are sufficiently different that we discuss them in separate chapters.

In this chapter we begin with a short comparison of classical and Bayesian paradigms, and then discuss the three main types of statistical inference from the classical point of view. The following chapter attempts to follow the same structure as this chapter, but from the Bayesian point of view. The topics covered in these two chapters complete the foundations for the remaining chapters on exploratory data analysis and data-based prediction methods.

4.1. Classical vs. Bayesian Statistical Inference

We start with a brief summary of the basic philosophical differences between the two approaches. A more detailed discussion of their practical pros and cons is presented in §5.9. We follow the balanced perspective set forth by Wasserman (Wass10; see §1.3), an expert in both Bayesian and frequentist theory.

Classical or frequentist statistics is based on these tenets:

- Probabilities refer to *relative frequencies of events*. They are objective properties of the real world.
- Parameters (such as the fraction of coin flips, for a certain coin, that are heads) are *fixed, unknown constants*. Because they are not fluctuating, probability statements about parameters are meaningless.
- Statistical procedures should have well-defined long-run frequency properties. For example, a 95% confidence interval should bracket the true value of the parameter with a limiting frequency of at least 95%.

In contrast, Bayesian inference takes this stance:

- Probability describes the degree of subjective belief, not the limiting frequency. Probability statements can be made about things *other than data*, including model parameters and models themselves.
- Inferences about a parameter are made by producing its probability distribution—*this distribution quantifies the uncertainty of our knowledge about that parameter*. Various point estimates, such as expectation value, may then be readily extracted from this distribution.

Note that both are equally concerned with uncertainties about estimates. The main difference is whether one is allowed, or not, to discuss the "probability" of some aspect of the fixed universe having a certain value. The choice between the two paradigms is in some sense a philosophical subtlety, but also has very real practical consequences. In terms of philosophy, there is a certain symmetry and thus elegance in the Bayesian construction giving a "grand unified theory" feel to it that appeals naturally to the sensibilities of many scientists.

In terms of pragmatism, the Bayesian approach is accompanied by significant difficulties, particularly when it comes to computation. Despite much advancement in the past few decades, these challenges should not be underestimated. In addition, the results of classical statistics are often the same as Bayesian results, and still represent a standard for reporting scientific results. Therefore, despite many strong advantages of the Bayesian approach, classical statistical procedures cannot be simply and fully replaced by Bayesian results. Indeed, maximum likelihood analysis, described next, is a major concept in both paradigms.

4.2. Maximum Likelihood Estimation (MLE)

We will start with *maximum likelihood estimation* (MLE), a common special case of the larger frequentist world. It is a good starting point because Bayesian estimation in the next chapter builds directly on top of the apparatus of maximum likelihood. The frequentist world, however, is not tied to the likelihood (as Bayesian estimation is), and we will also visit additional ideas outside of likelihood-based approaches later in this section (§4.2.8).

4.2.1. The Likelihood Function

The starting point for both MLE and Bayesian estimation is the *likelihood* of the data. The data likelihood represents a quantitative description of our measuring process. The concept was introduced by Gauss and Laplace, and popularized by Fisher in the first half of the last century.[1]

Given a known, or assumed, behavior of our measuring apparatus (or, statistically speaking, the distribution from which our sample was drawn), we can compute the probability, or likelihood, of observing any given value. For example, if we know or assume that our data $\{x_i\}$ are drawn from an $\mathcal{N}(\mu, \sigma)$ parent distribution, then the likelihood of a given value x_i is given by eq. 1.4. Assuming that individual values are independent (e.g., they would not be independent in the case of drawing from a small parent sample without replacement, such as the second drawing from an urn that initially contains a red and a white ball), the likelihood of the entire data set, L, is then the product of likelihoods for each particular value,

$$L \equiv p(\{x_i\}|M(\boldsymbol{\theta})) = \prod_{i=1}^{n} p(x_i|M(\boldsymbol{\theta})), \tag{4.1}$$

where M stands for a model (i.e., our understanding of the measurement process, or assumptions about it). In general, the model M includes k model parameters θ_p, $p = 1, \ldots, k$, abbreviated as the vector $\boldsymbol{\theta}$ with components θ_p. We will sometimes write just M, or just $\boldsymbol{\theta}$ and θ in one-dimensional cases, in place of $M(\boldsymbol{\theta})$, depending on what we are trying to emphasize. Instead of the specific one-dimensional data set $\{x_i\}$, we will often use D for data in general cases.

Although $L \equiv p(\{x_i\}|M(\boldsymbol{\theta}))$ can be read as "the probability of the data given the model," note that L is not a true (properly normalized) pdf. The likelihood of a given single value x_i is given by a true pdf (e.g., by eq. 1.4), but the product of such functions is no longer normalized to 1. The data likelihood L can take on extremely small values when a data set is large, and its logarithm is often used instead in practice, as we will see in the practical examples below.

The likelihood can be considered both as a function of the data and as a function of the model. When computing the likelihood of some data value x, L is a function of x for some fixed model parameters. Given some fixed data set, it can be considered as a function of the model parameters instead. These parameters can then be varied to maximize the likelihood of observing this specific data set, as described next. Note that this concept of likelihood maximization does *not* imply that likelihoods can be interpreted as probabilities for parameters $\boldsymbol{\theta}$. To do so, a full Bayesian analysis is required, as discussed in the next chapter.

4.2.2. The Maximum Likelihood Approach

Maximum likelihood estimation consists of the following conceptual steps:

1. The formulation of the data likelihood for some model M, $p(D|M)$, which amounts to an assumption about how the data are generated. This step is

[1] Fisher's first paper on this subject, and his first mathematical paper ever, was published in 1912 while he was still an undergraduate student.

crucial and the accuracy of the resulting inferences is strongly affected by the quality of this assumption (i.e., how well our model describes the actual data generation process). Models are typically described using a set of model parameters, $\boldsymbol{\theta}$, i.e., the model is $M(\boldsymbol{\theta})$.

2. Search for the best model parameters ($\boldsymbol{\theta}$) which maximize $p(D|M)$. This search yields the MLE *point estimates*, $\boldsymbol{\theta^0}$ (i.e., we obtain k estimates, θ_p^0, $p = 1, \ldots, k$).
3. Determination of the confidence region for model parameters, $\boldsymbol{\theta^0}$. Such a *confidence estimate* can be obtained analytically in MLE by doing mathematical derivations specific to the model chosen, but can also be done numerically for arbitrary models using general frequentist techniques, such as bootstrap, jackknife, and cross-validation, described later. Since the bootstrap can simulate draws of samples from the true underlying distribution of the data, various descriptive statistics can be computed on such samples to examine the uncertainties surrounding the data and our estimators based on that data.
4. Perform *hypothesis tests* as needed to make other conclusions about models and point estimates.

While these steps represent a blueprint for the frequentist approach in general, the likelihood is just one of many possible so-called objective functions (also called fitness functions, or cost functions); other possibilities are explored briefly in §4.2.8. An example of how to perform MLE in practice is described next, using a Gaussian likelihood.

4.2.3. The MLE Applied to a Homoscedastic Gaussian Likelihood

We will now solve a simple problem where we have a set of N measurements, $\{x_i\}$, of, say, the length of a rod. The measurement errors are known to be Gaussian, and we will consider two cases: here we analyze a case where all measurements have the same known error, σ (homoscedastic errors). In §4.2.6 we will analyze a case when each measurement has a different known error σ_i (heteroscedastic errors). The goal of our analysis in both cases is to find the maximum likelihood estimate for the length of the rod, and its confidence interval.

The likelihood for obtaining data $D = \{x_i\}$ given the rod length μ and measurement error σ is

$$L \equiv p(\{x_i\}|\mu, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x_i - \mu)^2}{2\sigma}\right). \tag{4.2}$$

Here there is only one model parameter, that is, $k = 1$ and $\theta_1 = \mu$. We can find the maximum likelihood estimate, μ^0, as the value of μ that maximizes L, as follows.

The *log-likelihood function* is defined by $\ln L \equiv \ln[L(\boldsymbol{\theta})]$. Its maximum occurs at the same place as that of the likelihood function, and the same is true of the likelihood function times any constant. Thus we shall often ignore constants in the likelihood function and work with its logarithm. The value of the model

parameter μ that maximizes lnL can be determined using the condition

$$\left.\frac{d\,\mathrm{lnL}(\mu)}{d\mu}\right|_{\mu^0} \equiv 0. \tag{4.3}$$

For our Gaussian example, we get from eq. 4.2,

$$\mathrm{lnL}(\mu) = \text{constant} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma^2}. \tag{4.4}$$

This particularly simple result is a direct consequence of the Gaussian error distribution and admits an analytic solution for MLE of μ derived from eq. 4.3,

$$\mu^0 = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{4.5}$$

That is, μ^0 is simply the arithmetic mean of all measurements (cf. eq. 3.31).

We started this example by assuming that σ is a measurement error and that there is a unique value of the length of a rod that we are trying to estimate. An alternative interpretation is that there is an intrinsic Gaussian distribution of rod lengths, given by σ, and the measurement errors are negligible compared to σ. We can combine these two possibilities by invoking convolution results for Gaussians as long as measurement errors are homoscedastic. Using eq. 3.45, we can interpret σ as the intrinsic distribution width and the measurement error added in quadrature. This simple result is valid only in the case of homoscedastic errors and the generalization to heteroscedastic errors is discussed below.

4.2.4. Properties of Maximum Likelihood Estimators

Maximum likelihood estimators have several optimality properties, under certain assumptions. The critical assumption is that the data truly come from the specified model class (e.g., they really are drawn from a Gaussian, if that is the model used). Additional assumptions include some relatively mild *regularity conditions*, which amount to various smoothness conditions, certain derivatives existing, etc. (see Lup93 for detailed discussion).

Maximum likelihood estimators have the following properties:

- They are *consistent* estimators; that is, they can be proven to converge to the true parameter value as the number of data points increases.
- They are *asymptotically normal* estimators. The distribution of the parameter estimate, as the number of data points increases to infinity, approaches a normal distribution, centered at the MLE, with a certain spread. This spread can often be easily calculated and used as a confidence band around the estimate, as discussed below (see eq. 4.7).
- They asymptotically achieve the theoretical minimum possible variance, called the Cramér–Rao bound. In other words, they achieve the best possible error given the data at hand; that is, no other estimator can do better in terms of efficiently using each data point to reduce the total error of the estimate (see eq. 3.33).

4.2.5. The MLE Confidence Intervals

Given an MLE, such as μ^0 above, how do we determine its uncertainty? The asymptotic normality of MLE is invoked to demonstrate that the error matrix can be computed from the covariance matrix as

$$\sigma_{jk} = \left(- \left. \frac{d^2 \ln \mathrm{L}}{d\theta_j \, d\theta_k} \right|_{\theta=\theta_0} \right)^{-1/2}. \tag{4.6}$$

This result is derived by expanding lnL in a Taylor series and retaining terms up to second order (essentially, lnL is approximated by a parabola, or an ellipsoidal surface in multidimensional cases, around its maximum). If this expansion is exact (as is the case for a Gaussian error distribution, see below), then eq. 4.6 is exact. In general, this is not the case and the likelihood surface can significantly deviate from a smooth elliptical surface. Furthermore, it often happens in practice that the likelihood surface is multimodal. It is always a good idea to visualize the likelihood surface when in doubt (see examples in §5.6).

The above expression is related to *expected Fisher information*, which is defined as the expectation value of the second derivative of −lnL with respect to θ (or the Fisher information matrix when θ is a vector). The inverse of the Fisher information gives a lower bound on the variance of any unbiased estimator of θ; this limit is known as the Cramér–Rao lower bound (for detailed discussion, see [9]). When evaluated at θ_0, such as in eq. 4.6, it is called *observed Fisher information* (for large samples, expected and observed Fisher information become asymptotically equal). Detailed discussion of the connection between Fisher information and confidence intervals, including asymptotic normality properties, is available in Wass10.

The diagonal elements, σ_{ii}, correspond to marginal error bars for parameters θ_i (in analogy with eq. 3.8). If $\sigma_{jk} = 0$ for $j \neq k$, then the inferred values of parameters are uncorrelated (i.e., the error in one parameter does not have an effect on other parameters; if somehow we were given the true value of one of the parameters, our estimates of the remaining parameters would not improve). In this case, the σ_{ii} are the direct analogs of error bars in one-dimensional problems.

It often happens that $\sigma_{jk} \neq 0$ when $j \neq k$. In this case, errors for parameters θ_j and θ_k are correlated (e.g., in the two-dimensional case, the likelihood surface is a bivariate Gaussian with principal axes that are not aligned with coordinate axes; see §3.5.2). This correlation tells us that some combinations of parameters are better determined than others. We can use eq. 3.84 for a two-dimensional case, and the results from §3.5.4 for a general case, to compute these parameter combinations (i.e., the principal axes of the error ellipse) and their uncertainties. When all diagonal elements are much larger than the off-diagonal elements, some combinations of parameters may be measured with a better accuracy than individual parameters.

Returning to our Gaussian example, the uncertainty of the mean is

$$\sigma_\mu = \left(- \left. \frac{d^2 \ln \mathrm{L}(\mu)}{d\mu^2} \right|_{\mu^0} \right)^{-1/2} = \frac{\sigma}{\sqrt{N}}, \tag{4.7}$$

which is the same as eq. 3.34. Note again that σ in this example was assumed to be constant and known.

4.2.6. The MLE Applied to a Heteroscedastic Gaussian Likelihood

The computation in the heteroscedastic case proceeds analogously to that in the homoscedastic case. The log-likelihood is

$$\ln L = \text{constant} - \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{2\sigma_i^2} \tag{4.8}$$

and again we can derive an analytic solution for the maximum likelihood estimator of μ:

$$\mu^0 = \frac{\sum_i^N w_i x_i}{\sum_i^N w_i}, \tag{4.9}$$

with weights $w_i = \sigma_i^{-2}$. That is, μ^0 is simply a weighted arithmetic mean of all measurements. When all σ_i are equal, eq. 4.9 reduces to the standard result given by eq. 3.31 and eq. 4.5.

Using eq. 4.6, the uncertainty of μ^0 is

$$\sigma_\mu = \left(\sum_{i=1}^{N} \frac{1}{\sigma_i^2} \right)^{-1/2} = \left(\sum_{i=1}^{N} w_i \right)^{-1/2}. \tag{4.10}$$

Again, when all σ_i are equal to σ, this reduces to the standard result given by eqs. 3.34 and 4.7.

We can generalize these results to the case when the measured quantity follows an intrinsic Gaussian distribution with *known* width σ_o, and measurement errors are also known and given as e_i. In this case, eq. 3.45 tells us that we can use the above results with $\sigma_i = (\sigma_o^2 + e_i^2)^{1/2}$. We will discuss important differences in cases when σ_o is not known when analyzing examples of Bayesian analysis in §5.6.1. Those examples will also shed more light on how to treat multidimensional likelihoods, as well as complex problems which need to be solved using numerical techniques.

4.2.7. The MLE in the Case of Truncated and Censored Data

Often the measuring apparatus does not sample the measured range of variable x with equal probability. The probability of drawing a measurement x is quantified using the selection probability, or selection function, $S(x)$. When $S(x) = 0$ for $x > x_{\rm max}$ (analogously for $x < x_{\rm min}$), the data set is *truncated* and we know nothing about sources with $x > x_{\rm max}$ (not even whether they exist or not). A related but different concept is *censored* data sets, where a measurement of an *existing* source was attempted, but the value is outside of some known interval (a familiar astronomical case is an "upper limit" for flux measurement when we look for, e.g., an X-ray source in an optical image of the same region on the sky but do not find it).

We will now revisit the Gaussian example and discuss how to account for data truncation using the MLE approach. For simplicity, we will assume that the selection function is unity for $x_{\rm min} \leq x \leq x_{\rm max}$ and $S(x)$=0 otherwise. The treatment of a more

complicated selection function is discussed in §4.9, and censored data are discussed in the context of regression in §8.1.

The key point when accounting for truncated data is that the data likelihood of a single datum must be a properly normalized pdf. The fact that data are truncated enters analysis through a renormalization constant. In the case of a Gaussian error distribution (we assume that σ is known), the likelihood for a single data point is

$$p(x_i|\mu, \sigma, x_{\rm min}, x_{\rm max}) = C(\mu, x_{\rm min}, x_{\rm max}) \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-(x_i-\mu)^2}{2\sigma}\right), \qquad (4.11)$$

where the renormalization constant is evaluated as

$$C(\mu, \sigma, x_{\rm min}, x_{\rm max}) = (P(x_{\rm max}|\mu, \sigma) - P(x_{\rm min}|\mu, \sigma))^{-1} \qquad (4.12)$$

with the cumulative distribution function for Gaussian, P, given by eq. 3.48.

The log-likelihood is

$$\ln L(\mu) = \text{constant} - \sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma^2} + N \ln\left[C(\mu, \sigma, x_{\rm min}, x_{\rm max})\right]. \qquad (4.13)$$

The first two terms on the right-hand side are identical to those in eq. 4.4, and the third term accounts for truncation. Note that the third term does not depend on data because $x_{\rm min}$ and $x_{\rm max}$ are the same for all points; it would be straightforward to incorporate varying $x_{\rm min}$ and $x_{\rm max}$ (i.e., data points with different "selection limits"). Because the Gauss error function (see eq. 3.47) is needed to evaluate the renormalization constant, there is no simple closed-form expression for μ^0 in this case.

For an illustrative numerical example, let us consider a heavily truncated Gaussian and the following estimation problem. In the country Utopia, graduate schools are so good that students from the country of Karpathia prefer to study in Utopia. They are admitted to a graduate school if they pass a number of tests, which we will assume leads to a hard lower limit of 120 for the IQ score of Karpathian students. We also know that IQ follows a normal distribution centered at 100 with a standard deviation of 15. If you meet a Karpathian student with an IQ of $S_{\rm IQ}$, what is your best estimate for the mean IQ of Karpathia?

The answer can be easily obtained by finding the maximum of $\ln L(\mu)$ given by eq. 4.13, evaluated for $x_{\rm min} = 120$, $x_{\rm max} = \infty$, $N = 1$, $x_1 = S_{\rm IQ}$, and $\sigma = 15$. Whether the MLE for μ, μ^0, is larger or smaller than 100 depends on the exact value of $S_{\rm IQ}$. It is easy to show (e.g., using a mock sample) that the mean value for an $\mathcal{N}(100, 15)$ Gaussian truncated at $x_{\rm min} = 120$ is $\sim$127. If $S_{\rm IQ}$ is smaller than 127, the implied mean IQ for Karpathia is smaller than 100, and conversely if $S_{\rm IQ} > 127$, then $\mu^0 > 100$. For example, if $S_{\rm IQ} = 140$, the MLE for μ is 130. For a sample of two students with an IQ of 120 and 140, $\mu^0 = 107$, with uncertainty $\sigma_\mu = 20$. For an arbitrary number of students, their mean IQ must be greater than 127 to obtain $\mu^0 > 100$, and the sample size must be considerable to, for example, reject the hypothesis (see §4.6) that the mean IQ of Karpathia is smaller (or larger) than 100. If all your Karpathian friends have an IQ around 120, bunched next to the selection

threshold, it is likely that the mean IQ in Karpathia is below 100! Therefore, if you run into a smart Karpathian, do not automatically assume that all Karpathians have high IQs on average because it could be due to selection effects. Note that if you had a large sample of Karpathian students, you could bin their IQ scores and fit a Gaussian (the data would only constrain the tail of the Gaussian). Such regression methods are discussed in chapter 8. However, as this example shows, there is no need to bin your data, except perhaps for visualization purposes.

4.2.8. Beyond the Likelihood: Other Cost Functions and Robustness

Maximum likelihood represents perhaps the most common choice of the so-called "cost function" (or objective function) within the frequentist paradigm, but not the only one. Here the cost function quantifies some "cost" associated with parameter estimation. The expectation value of the cost function is called "risk" and can be minimized to obtain best-fit parameters.

The mean integrated square error (MISE), defined as

$$\mathrm{MISE} = \int_{-\infty}^{+\infty} [f(x) - h(x)]^2 \, dx, \tag{4.14}$$

is an often-used form of risk; it shows how "close" is our empirical estimate $f(x)$ to the true pdf $h(x)$. The MISE is based on a cost function given by the mean square error, also known as the L_2 norm. A cost function that minimizes absolute deviation is called the L_1 norm. As shown in examples earlier in this section, the MLE applied to a Gaussian likelihood leads to an L_2 cost function (see eq. 4.4). If data instead followed the Laplace (exponential) distribution (see §3.3.6), the MLE would yield an L_1 cost function.

There are many other possible cost functions and often they represent a distinctive feature of a given algorithm. Some cost functions are specifically designed to be robust to outliers, and can thus be useful when analyzing contaminated data (see §8.9 for some examples). The concept of a cost function is especially important in cases where it is hard to formalize the likelihood function, because an optimal solution can still be found by minimizing the corresponding risk. We will address cost functions in more detail when discussing various methods in chapters 6–10.

4.3. The Goodness of Fit and Model Selection

When using maximum likelihood methods, the MLE approach estimates the "best-fit" model parameters and gives us their uncertainties, but it does not tell us how good the fit is. For example, the results given in §4.2.3 and §4.2.6 will tell us the best-fit parameters of a Gaussian, but what if our data was not drawn from a Gaussian distribution? If we select another model, say a Laplace distribution, how do we compare the two possibilities? This comparison becomes even more involved when models have a varying number of model parameters. For example, we know that a fifth-order polynomial fit will always be a better fit to data than a straight-line fit, but do the data really support such a sophisticated model?

4.3.1. The Goodness of Fit for a Model

Using the best-fit parameters, we can compute the maximum value of the likelihood from eq. 4.1, which we will call L^0. Assuming that our model is correct, we can ask how likely it is that this particular value would have arisen by chance. If it is very unlikely to obtain L^0, or $\ln L^0$, by randomly drawing data from the implied best-fit distribution, then the best-fit model is not a good description of the data. Evidently, we need to be able to predict the distribution of L, or equivalently lnL.

For the case of the Gaussian likelihood, we can rewrite eq. 4.4 as

$$\ln L = \text{constant} - \frac{1}{2}\sum_{i=1}^{N} z_i^2 = \text{constant} - \frac{1}{2}\chi^2, \tag{4.15}$$

where $z_i = (x_i - \mu)/\sigma$. Therefore, the distribution of lnL can be determined from the χ^2 distribution with $N - k$ degrees of freedom (see §3.3.7), where k is the number of model parameters determined from data (in this example $k = 1$ because μ is determined from data and σ was assumed fixed). The distribution of χ^2 does not depend on the actual values of μ and σ; the expectation value for the χ^2 distribution is $N - k$ and its standard deviation is $\sqrt{2(N-k)}$. For a "good fit," we expect that χ^2 *per degree of freedom*,

$$\chi^2_{\rm dof} = \frac{1}{N-k}\sum_{i=1}^{N} z_i^2 \approx 1. \tag{4.16}$$

If instead $(\chi^2_{\rm dof} - 1)$ is many times larger than $\sqrt{2/(N-k)}$, it is unlikely that the data were generated by the assumed model. Note, however, that outliers may significantly increase $\chi^2_{\rm dof}$. The likelihood of a particular value of $\chi^2_{\rm dof}$ for a given number of degrees of freedom can be found in tables or evaluated using the function `scipy.stats.chi2`.

As an example, consider the simple case of the luminosity of a single star being measured multiple times (figure 4.1). Our model is that of a star with no intrinsic luminosity variation. If the model and measurement errors are consistent, this will lead to $\chi^2_{\rm dof}$ close to 1. Overestimating the measurement errors can lead to an improbably low $\chi^2_{\rm dof}$, while underestimating the measurement errors can lead to an improbably high $\chi^2_{\rm dof}$. A high $\chi^2_{\rm dof}$ may also indicate that the model is insufficient to fit the data: for example, if the star has intrinsic variation which is either periodic (e.g., in the so-called RR-Lyrae-type variable stars) or stochastic (e.g., active M dwarf stars). In this case, accounting for this variability in the model can lead to a better fit to the data. We will explore these options in later chapters. Because the number of samples is large ($N = 50$), the χ^2 distribution is approximately Gaussian: to aid in evaluating the fits, figure 4.1 reports the deviation in σ for each fit.

The probability that a certain maximum likelihood value L^0 might have arisen by chance can be evaluated using the χ^2 distribution only when the likelihood is Gaussian. When the likelihood is not Gaussian (e.g., when analyzing small count data which follows the Poisson distribution), L^0 is still a measure of how well a model fits the data. Different models, assuming that they have the same number of free parameters, can be ranked in terms of L^0. For example, we could derive the best-fit estimates of a Laplace distribution using MLE, and compare the resulting L^0 to the value obtained for a Gaussian distribution.

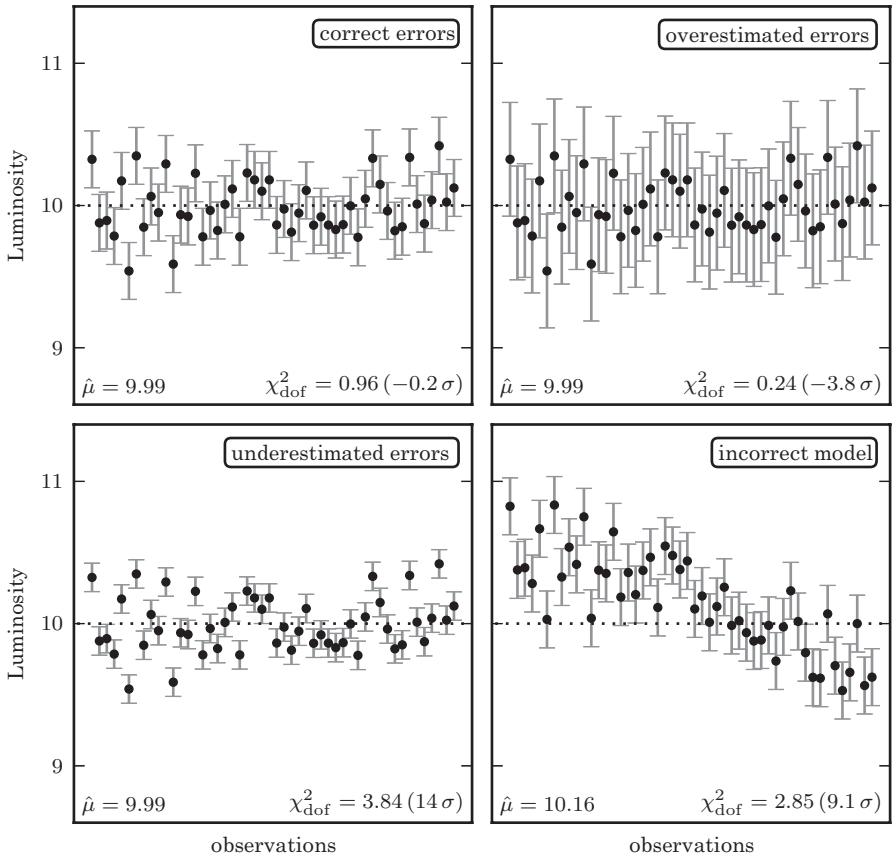

Figure 4.1. The use of the χ^2 statistic for evaluating the goodness of fit. The data here are a series of observations of the luminosity of a star, with known error bars. Our model assumes that the brightness of the star does not vary; that is, all the scatter in the data is due to measurement error. $\chi^2_{\rm dof} \approx 1$ indicates that the model fits the data well (upper-left panel). $\chi^2_{\rm dof}$ much smaller than 1 (upper-right panel) is an indication that the errors are overestimated. $\chi^2_{\rm dof}$ much larger than 1 is an indication either that the errors are underestimated (lower-left panel) or that the model is not a good description of the data (lower-right panel). In this last case, it is clear from the data that the star's luminosity is varying with time: this situation will be treated more fully in chapter 10.

Note, however, that L^0 by itself does not tell us how well a model fits the data. That is, we do not know in general if a particular value of L^0 is consistent with simply arising by chance, as opposed to a model being inadequate. To quantify this probability, we need to know the expected distribution of L^0, as given by the χ^2 distribution in the special case of Gaussian likelihood.

4.3.2. Model Comparison

Given the maximum likelihood for a set of models, $L^0(M)$, the model with the largest value provides the best description of the data. However, this is not necessarily the best model overall when models have different numbers of free parameters.

A "scoring" system also needs to take into account the model complexity and "penalize" models for additional parameters not supported by data. In the Bayesian framework, there is a unique scoring system based on the posterior model probability, as discussed in detail in §5.4.3. Here we limit discussion to classical methods.

There are several methods for comparing models in classical statistics. For example, we discuss the cross-validation technique and bias–variance trade-off (based on mean square error, as discussed above in the context of cost functions) in the context of regression in chapter 8. A popular general classical method for model comparison is the Aikake information criterion (AIC). The AIC is a simple approach based on an asymptotic approximation; the preferred approach to model comparison for the highest accuracy is cross-validation (discussed in §8.11.1), which is based on only the finite data at hand rather than approximations based on infinite data. Nonetheless the AIC is easy to use, and often effective for simple models.

The AIC (the version corrected for small samples, see [15]) is computed as

$$\mathrm{AIC} \equiv -2\ln\left(L^0(M)\right) + 2k + \frac{2k(k+1)}{N-k-1}, \tag{4.17}$$

where k is the number of model parameters and N is the number of data points. The AIC is related to methods based on bias–variance trade-off (see Wass10), and can be derived using information theory (see HTF09).

Under the assumption of normality, the first term is equal to the model's χ^2 (up to a constant). When multiple models are compared, the one with the smallest AIC is the best model to select. If the models are equally successful in describing the data (they have the same value of $L^0(M)$), then the model with fewer free parameters wins. A closely related concept to AIC is the Bayesian information criterion (BIC), introduced in §5.4.3. We will discuss an example of model selection using AIC and BIC criteria in the following section.

4.4. ML Applied to Gaussian Mixtures: The Expectation Maximization Algorithm

Data likelihood can be a complex function of many parameters that often does not admit an easy analytic solution for MLE. In such cases, numerical methods such as those described in §5.8, are used to obtain model parameters and their uncertainties.

A special case of a fairly complex likelihood which can still be maximized using a relatively simple and straightforward numerical method is a mixture of Gaussians. We first describe the model and the resulting data likelihood function, and then discuss the expectation maximization algorithm for maximizing the likelihood.

4.4.1. Gaussian Mixture Model

The likelihood of a datum x_i for a Gaussian mixture model is given by

$$p(x_i|\boldsymbol{\theta}) = \sum_{j=1}^{M} \alpha_j \mathcal{N}(\mu_j, \sigma_j), \tag{4.18}$$

where dependence on x_i comes via a Gaussian $\mathcal{N}(\mu_j, \sigma_j)$. The vector of parameters $\boldsymbol{\theta}$ that need to be estimated for a given data set $\{x_i\}$ includes normalization factors for each Gaussian, α_j, and its parameters μ_j and σ_j. It is assumed that the data have negligible uncertainties (e.g., compared to the smallest σ_j), and that M is given. We shall see below how to relax both of these assumptions. Given that the likelihood for a single datum must be a true pdf, α_j must satisfy the normalization constraint

$$\sum_{j=1}^{M} \alpha_j = 1. \tag{4.19}$$

The log-likelihood for the whole data set is then

$$\ln L = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{M} \alpha_j \mathcal{N}(\mu_j, \sigma_j) \right] \tag{4.20}$$

and needs to be maximized as a function of $k = (3M - 1)$ parameters (not $3M$ because of the constraint given by eq. 4.19).

An attempt to derive constraints on these parameters by setting partial derivatives of lnL with respect to each parameter to zero would result in a complex system of $(3M - 1)$ nonlinear equations and would not bring us much closer to the solution (the problem is that in eq. 4.20 the logarithm is taken of the whole sum over j classes, unlike in the case of a single Gaussian where taking the logarithm of the exponential function results in a simple quadratic function). We could also attempt to simply find the maximum of lnL through an exhaustive search of the parameter space. However, even when M is small, such an exhaustive search would be too time consuming, and for large M it becomes impossible. For example, if the search grid for each parameter included only 10 values (typically insufficient to achieve the required parameter accuracy), even with a relatively small $M = 5$, we would have to evaluate the function given by eq. 4.20 about 10^{14} times!

A practical solution for maximizing lnL is to use the Levenberg–Marquardt algorithm, which combines gradient descent and Gauss–Newton optimization (see NumRec). Another possibility is to use Markov chain Monte Carlo methods, discussed in detail in §5.8. However, a much faster procedure is available, especially for the case of large M, based on the concept of hidden variables, as described in the next section.

4.4.2. Class Labels and Hidden Variables

The likelihood given by eq. 4.18 can be interpreted using the concept of "hidden" (or missing) variables. If M Gaussian components are interpreted as different "classes," which means that a particular datum x_i was generated by one and only one of the individual Gaussian components, then the index j is called a "class label." The hidden variable here is the class label j responsible for generating each x_i. If we knew the class label for each datum, then this maximization problem would be trivial and equivalent to examples based on a single Gaussian distribution discussed in the previous section. That is, all the data could be sorted into M subsamples according to their class label. The fraction of points in each subsample would be an estimator

of α_j, while μ_j and σ_j could be trivially obtained using eqs. 3.31 and 3.32. In a more general case when the probability function for each class is described by a non-Gaussian function, eqs. 3.31 and 3.32 cannot be used, but given that we know the class labels the problem can still be solved and corresponds to the so-called naive Bayesian classifier discussed in §9.3.2.

Since the class labels are not known, for each data value we can only determine the probability that it was generated by class j (sometimes called responsibility, e.g., HTF09). Given x_i, this probability can be obtained for each class using Bayes' rule (see eq. 3.10),

$$p(j|x_i) = \frac{\alpha_j \mathcal{N}(\mu_j, \sigma_j)}{\sum_{j=1}^{M} \alpha_j \mathcal{N}(\mu_j, \sigma_j)}. \tag{4.21}$$

The class probability $p(j|x_i)$ is small when x_i is not within "a few" σ_j from μ_j (assuming that x_i is close to some other mixture component). Of course, $\sum_{j=1}^{M} p(j|x_i) = 1$. This probabilistic class assignment is directly related to the hypothesis testing concept introduced in §4.6, and will be discussed in more detail in chapter 9.

4.4.3. The Basics of the Expectation Maximization Algorithm

Of course, we do not have to interpret eq. 4.18 in terms of classes and hidden variables. After all, lnL is just a scalar function that needs to be maximized. However, this interpretation leads to an algorithm, called the expectation maximization (EM) algorithm, which can be used to make this maximization fast and straightforward in practice. The EM algorithm was introduced by Dempster, Laird, and Rubin in 1977 ([7]), and since then many books have been written about its various aspects (for a good short tutorial, see [20]).

The key ingredient of the iterative EM algorithm is the assumption that the class probability $p(j|x_i)$ is known and fixed in each iteration (for a justification based on conditional probabilities, see [20] and HTF09). The EM algorithm is not limited to Gaussian mixtures, so instead of $\mathcal{N}(\mu_j, \sigma_j)$ in eq. 4.18, let us use a more general pdf for each component, $p_j(x_i|\boldsymbol{\theta})$ (for notational simplicity, we do not explicitly account for the fact that p_j includes only a subset of all $\boldsymbol{\theta}$ parameters, e.g., only μ_j and σ_j are relevant for the jth Gaussian component). By analogy with eq. 4.20, the log-likelihood is

$$\mathrm{lnL} = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{M} \alpha_j \, p_j(x_i|\boldsymbol{\theta}) \right]. \tag{4.22}$$

We can take a partial derivative of lnL with respect to the parameter θ_j,

$$\frac{\partial \mathrm{lnL}}{\partial \theta_j} = \sum_{i=1}^{N} \frac{\alpha_j}{\sum_{j=1}^{M} \alpha_j \, p_j(x_i|\boldsymbol{\theta})} \left[\frac{\partial p_j(x_i|\boldsymbol{\theta})}{\partial \theta_j} \right] \tag{4.23}$$

and motivated by eq. 4.21, rewrite it as

$$\frac{\partial \ln \mathrm{L}}{\partial \theta_j} = \sum_{i=1}^{N} \left[\frac{\alpha_j \, p_j(x_i|\boldsymbol{\theta})}{\sum_{j=1}^{M} \alpha_j \, p_j(x_i|\boldsymbol{\theta})} \right] \left[\frac{1}{p_j(x_i|\boldsymbol{\theta})} \frac{\partial p_j(x_i|\boldsymbol{\theta})}{\partial \theta_j} \right]. \tag{4.24}$$

Although this equation looks horrendous, it can be greatly simplified. The first term corresponds to the class probability given by eq. 4.21. Because it will be fixed in a given iteration, we introduce a shorthand $w_{ij} = p(j|x_i)$. The second term is the partial derivative of $\ln[p_j(x_i|\boldsymbol{\theta})]$. When $p_j(x_i|\boldsymbol{\theta})$ is Gaussian, it leads to particularly simple constraints for model parameters because now we take the logarithm of the exponential function *before* taking the derivative. Therefore,

$$\frac{\partial \ln \mathrm{L}}{\partial \theta_j} = -\sum_{i=1}^{N} w_{ij} \frac{\partial}{\partial \theta_j} \left[\ln \sigma_j + \frac{(x_i - \mu_j)^2}{2\,\sigma_j^2} \right], \tag{4.25}$$

where θ_j now corresponds to μ_j or σ_j. By setting the derivatives of lnL with respect to μ_j and σ_j to zero, we get the estimators (this derivation is discussed in more detail in §5.6.1)

$$\mu_j = \frac{\sum_{i=1}^{N} w_{ij} x_i}{\sum_{i=1}^{N} w_{ij}}, \tag{4.26}$$

$$\sigma_j^2 = \frac{\sum_{i=1}^{N} w_{ij} (x_i - \mu_j)^2}{\sum_{i=1}^{N} w_{ij}}, \tag{4.27}$$

and from the normalization constraint,

$$\alpha_j = \frac{1}{N} \sum_{i=1}^{N} w_{ij}. \tag{4.28}$$

These expressions and eq. 4.21 form the basis of the iterative EM algorithm in the case of Gaussian mixtures. Starting with a guess for w_{ij}, the values of α_j, μ_j, and σ_j are estimated using eqs. 4.26–4.28. This is the "maximization" *M-step* which brings the parameters closer toward the local maximum. In the subsequent "expectation" *E-step*, w_{ij} are updated using eq. 4.21. The algorithm is not sensitive to the initial guess of parameter values. For example, setting all σ_j to the sample standard deviation, all α_j to $1/M$, and randomly drawing μ_j from the observed $\{x_i\}$ values, typically works well in practice (see HTF09).

Treating w_{ij} as constants during the M-step may sound ad hoc, and the whole EM algorithm might look like a heuristic method. After all, the above derivation does not guarantee that the algorithm will converge. Nevertheless, the EM algorithm has a rigorous foundation and it is provable that it will indeed find a local maximum of lnL for a wide class of likelihood functions (for discussion and references, see [20, 25]). In practice, however, the EM algorithm may fail due to numerical difficulties, especially

when the available data are sparsely distributed, in the case of outliers, and if some data points are repeated (see [1]).

Scikit-learn contains an EM algorithm for fitting N-dimensional mixtures of Gaussians:

```
>>> import numpy as np
>>> from sklearn.mixture import GMM
>>> X = np.random.normal(size=(100, 1)) # 100 points
        # in 1 dim
>>> model = GMM(2) # two components
>>> model.fit(X)
>>> model.means_ # the locations of the best-fit
    # components
array([[-0.05786756],
     [ 0.69668864]])
```

See the source code of figure 4.2 for a further example. Multidimensional Gaussian mixture models are also discussed in the context of clustering and density estimation: see §6.3.

How to choose the number of classes?

We have assumed in the above discussion of the EM algorithm that the number of classes in a mixture, M, is known. As M is increased, the description of the data set $\{x_i\}$ using a mixture model will steadily improve. On the other hand, a very large M is undesired—after all, $M = N$ will assign a mixture component to each point in a data set. How do we choose M in practice?

Selecting an optimal M for a mixture model is a case of model selection discussed in §4.3. Essentially, we evaluate multiple models and score them according to some metric to get the best M. Additional detailed discussion of this important topic from the Bayesian viewpoint is presented in §5.4. A basic example of this is shown in figure 4.2, where the AIC and BIC are used to choose the optimal number of components to represent a simulated data set generated using a mixture of three Gaussian distributions. Using these metrics, the correct optimal $M = 3$ is readily recognized.

The EM algorithm as a classification tool

The right panel in figure 4.2 shows the class probability for the optimal model ($M = 3$) as a function of x (cf. eq. 4.21). These results can be used to probabilistically assign all measured values $\{x_i\}$ to one of the three classes (mixture components). There is no unique way to deterministically assign a class to each of the data points because there are unknown hidden parameters. In practice, the so-called completeness vs. contamination trade-off plays a major role in selecting classification thresholds (for a detailed discussion see §4.6).

Results analogous to the example shown in figure 4.2 can be obtained in multidimensional cases, where the mixture involves multivariate Gaussian distributions

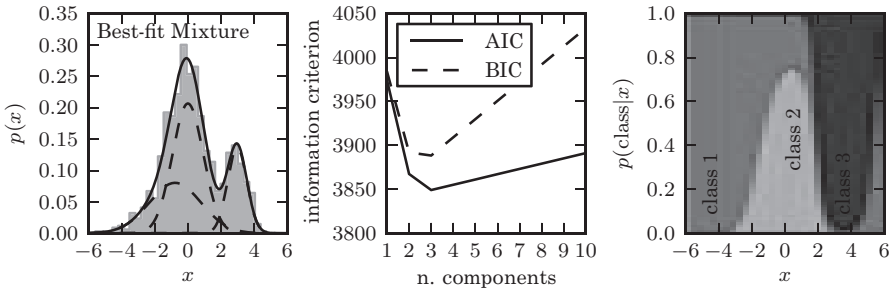

Figure 4.2. Example of a one-dimensional Gaussian mixture model with three components. The left panel shows a histogram of the data, along with the best-fit model for a mixture with three components. The center panel shows the model selection criteria AIC (see §4.3) and BIC (see §5.4) as a function of the number of components. Both are minimized for a three-component model. The right panel shows the probability that a given point is drawn from each class as a function of its position. For a given x value, the vertical extent of each region is proportional to that probability. Note that extreme values are most likely to belong to class 1.

discussed in §3.5.4. Here too an optimal model can be used to assign a probabilistic classification to each data point, and this and other classification methods are discussed in detail in chapter 9.

How to account for measurement errors?

In the above discussion of the EM algorithm, it was assumed that measurement errors for $\{x_i\}$ are negligible when compared to the smallest component width, σ_j. However, in practice this assumption is often not acceptable and the best-fit σ_j that are "broadened" by measurement errors are biased estimates of "intrinsic" widths (e.g., when measuring the widths of spectral lines). How can we account for errors in x_i, given as e_i?

We will limit our discussion to Gaussian mixtures, and assume that measurement uncertainties, as quantified by e_i, follow a Gaussian distribution. In the case of homoscedastic errors, where all $e_i = e$, we can make use of the fact that the convolution of two Gaussians is a Gaussian (see eq. 3.45) and obtain intrinsic widths as

$$\sigma_j^* = (\sigma_j^2 - e^2)^{1/2}. \tag{4.29}$$

This "poor-man's" correction procedure fails in the heteroscedastic case. Furthermore, due to uncertainties in the best-fit values, it is entirely possible that the best-fit value of σ_j may turn out to be smaller than e.

A remedy is to account for measurement errors already in the model description: we can replace σ_j in eq. 4.18 by $(\sigma_j^2 + e_i^2)^{1/2}$, where the σ_j now correspond to the intrinsic widths of each class. However, these new class pdfs do not admit simple explicit prescriptions for the maximization step given by eqs. 4.26–4.27 because they are no longer Gaussian (see §5.6.1 for a related discussion).

Following the same derivation steps, the new prescriptions for the M-step are now

$$\mu_j = \frac{\sum_{i=1}^{N} \frac{w_{ij}}{\sigma_j^2 + e_i^2} x_i}{\sum_{i=1}^{N} \frac{w_{ij}}{\sigma_j^2 + e_i^2}} \tag{4.30}$$

and

$$\sum_{i=1}^{N} \frac{w_{ij}}{\sigma_j^2 + e_i^2} = \sum_{i=1}^{N} \frac{w_{ij}}{(\sigma_j^2 + e_i^2)^2} (x_i - \mu_j)^2. \tag{4.31}$$

Compared to eqs. 4.26–4.27, σ_j is now "coupled" to e_i and cannot be moved outside of the sum, which prevents a few cancelations that led to simple forms when $e_i = 0$. Update rules for μ_j and α_j are still explicit and would require only a minor modification of specific implementation. The main difficulty, which prevents the use of standard EM routines for performing the M-step, is eq. 4.31, because the update rule for σ_j is not explicit anymore. Nevertheless, it still provides a rule for updating σ_j, which can be found by numerically solving eq. 4.31. We discuss a very similar problem and provide a numerical example in §5.6.1. An expectation maximization approach to Gaussian mixture models in the presence of errors is also discussed in chapter 6 (under the heading *extreme deconvolution* [5]) in the context of clustering and density estimation.

Non-Gaussian mixture models

The EM algorithm is not confined to Gaussian mixtures. As eq. 4.24 shows, the basic premise of the method can be derived for any mixture model. In addition to various useful properties discussed in §3.3.2, a major benefit of Gaussian pdfs is the very simple set of explicit equations (eqs. 4.26–4.28) for updating model parameters. When other pdfs are used, a variety of techniques are proposed in the literature for implementation of the maximization M-step. For cases where Gaussian mixtures are insufficient descriptors of data, we recommend consulting abundant and easily accessible literature on the various forms of the EM algorithm.

4.5. Confidence Estimates: the Bootstrap and the Jackknife

Most standard expressions for computing confidence limits for estimated parameters are based on fairly strong assumptions, such as Gaussianity and large samples. Fortunately, there are two alternative methods for computing confidence limits that are general, powerful, and easy to implement. Compared to the rest of statistics, they are relatively new and are made possible by the advent of cheap computing power. Both rely on resampling of the data set $\{x_i\}$.

Our data set $\{x_i\}$ is drawn from some distribution function $h(x)$. If we knew $h(x)$ perfectly well, we could compute any statistic without uncertainty (e.g., we could draw a large sample from $h(x)$ and, e.g., compute the mean). However, we do not know $h(x)$, and the best we can do are computations which rely on various estimates of $h(x)$ derived from the data, which we call here $f(x)$. Bootstrapping is based on the

approximation (see Lup93)

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} \delta(x - x_i), \tag{4.32}$$

where $\delta(x)$ is the Dirac δ function. The function $f(x)$ maximizes the probability of obtaining observed data values (f is a maximum likelihood estimator of h; for a discussion of bootstrap from the Bayesian viewpoint, see HTF09). We can now pretend that $f(x)$ is actually $h(x)$, and use it to perform various computations. For example, we could use eq. 4.32 to estimate the mean and its uncertainty.

When determining parameter uncertainties in practice, we use eq. 4.32 to draw an almost arbitrary number of new data sets. There are $N!$ possible distinct samples of size N and the probability that a new data set is identical to the original data set is $N!/N^n$ (even for a small N, this probability is small, e.g., for $N = 10$ it is only 0.00036). In other words, we draw from the observed data set with replacement: select N new index values j from the range $i = 1, \ldots, N$, and this is your new sample (some values from $\{x_i\}$ can appear twice or more times in the resampled data set). This resampling is done B times, and the resulting B data sets are used to compute the statistic of interest B times. The distribution of these values maps the uncertainty of the statistics of interest and can be used to estimate its bias and standard error, as well as other statistics.

The bootstrap method was proposed by Efron in 1979 [8]; more information can be found in [13] and references therein. The bootstrap method described above is called the nonparametric bootstrap; there is also the parametric bootstrap method which draws samples from the best-fit model. According to Wall and Jenkins, Efron named this method after "the image of lifting oneself up by one's own bootstraps." The nonparametric bootstrap method is especially useful when errors for individual data values are not independent (e.g., cumulative histogram, or two-point correlation function; see §6.5). Nevertheless, astronomers sometimes misuse the bootstrap idea (see [17]) and ignore nontrivial implementation considerations in complex problems; for example, the treatment in NumRec is misleadingly simple (for a detailed discussion of recent developments in bootstrap methodology and an excellent reference list, see [6]).

Figure 4.3 illustrates an application of the bootstrap method for estimating uncertainty in the standard deviation and σ_G (see eq. 3.36). The data sample has $N = 1000$ values and it was drawn from a Gaussian distribution with $\mu = 0$ and $\sigma = 1$. It was resampled 10,000 times and the histograms in figure 4.3 show the distribution of the resulting σ and σ_G. We can see that the bootstrap estimates of uncertainties are in good agreement with the values computed using eqs. 3.35 and 3.37.

The jackknife method, invented by Tukey in 1958, is similar in spirit to the bootstrap method (according to Lup93, the name jackknife implies robustness and general applicability). Rather than drawing a data set of the same size as the original data set during the resampling step, one or more observations are left unused when computing the statistic of interest. Let us call this statistic α with its value computed from the full data set α_N. Assuming that one observation (data value) is removed when resampling, we can form N such data sets, and compute a statistic of interest,

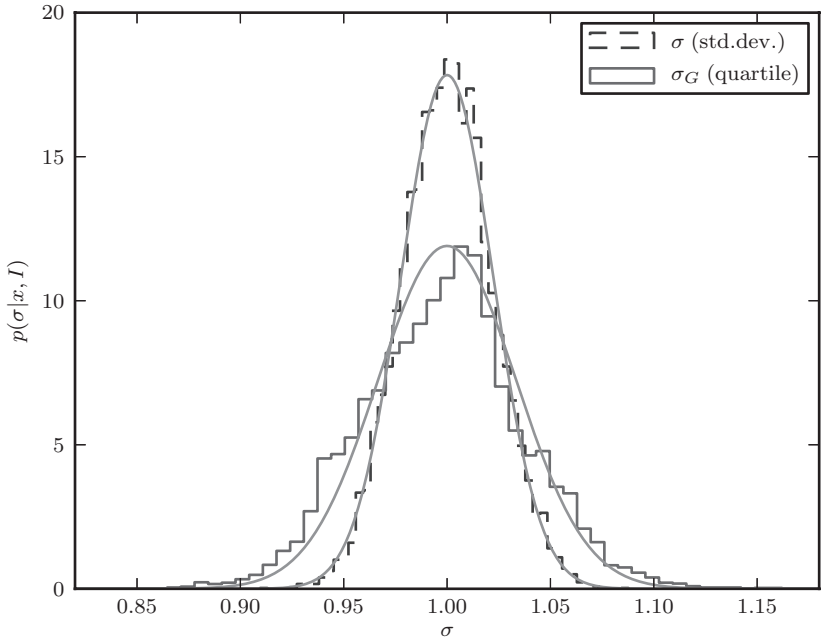

Figure 4.3. The bootstrap uncertainty estimates for the sample standard deviation σ (dashed line; see eq. 3.32) and σ_G (solid line; see eq. 3.36). The sample consists of $N = 1000$ values drawn from a Gaussian distribution with $\mu = 0$ and $\sigma = 1$. The bootstrap estimates are based on 10,000 samples. The thin lines show Gaussians with the widths determined as $s/\sqrt{2(N-1)}$ (eq. 3.35) for σ and $1.06s/\sqrt{N}$ (eq. 3.37) for σ_G.

α_i^*, for each of them. It can be shown that in the case of a single observation removed from the data set, a bias-corrected jackknife estimate of α can be computed as (see Lup93 for a concise derivation)

$$\alpha^J = \alpha_N + \Delta\alpha, \tag{4.33}$$

where the jackknife correction is

$$\Delta\alpha = (N-1)\left(\alpha_N - \frac{1}{N}\sum_{i=1}^{N}\alpha_i^*\right). \tag{4.34}$$

For estimators which are asymptotically normal, the standard error for a jackknife estimate α^J is

$$\sigma_\alpha = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}\left[N\alpha_N - \alpha^J - (N-1)\alpha_i^*\right]^2}. \tag{4.35}$$

The confidence limits for α can be computed using Student's t distribution (see §3.3.8) with $t = (\alpha - \alpha^J)/\sigma_\alpha$ and $N-1$ degrees of freedom. The jackknife standard error is more reliable than the jackknife bias correction because it is based on a

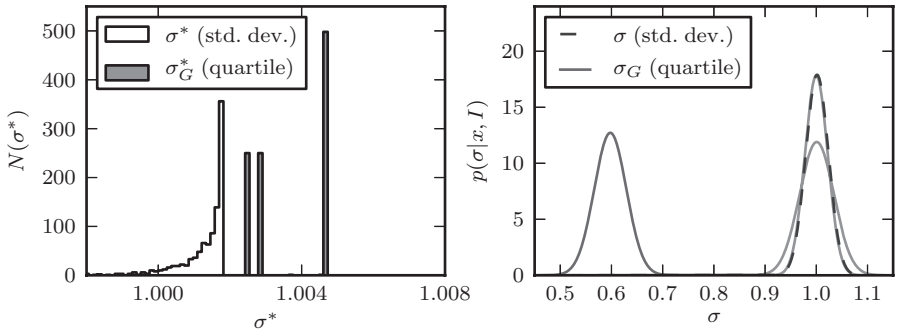

Figure 4.4. The jackknife uncertainty estimates for the width of a Gaussian distribution. This example uses the same data as figure 4.3. The upper panel shows a histogram of the widths determined using the sample standard deviation, and using the interquartile range. The lower panel shows the corrected jackknife estimates (eqs. 4.33 and 4.35) for the two methods. The gray lines show the theoretical results, given by eq. 3.35 for σ and eq. 3.37 for σ_G. The result for σ matches the theoretical result almost exactly, but note the failure of the jackknife to correctly estimate σ_G (see the text for a discussion of this result).

simpler approximation (see Lup93 for a detailed discussion). For the sample whose bootstrap uncertainty estimates for σ and σ_G are shown in figure 4.3, the jackknife method (eq. 4.35) gives similar widths as with the bootstrap method. Note, however, that the bias correction estimate for σ_G given by eq. 4.34 is completely unreliable (see figure 4.4). This failure is a general problem with the standard jackknife method, which performs well for smooth differential statistics such as the mean and standard deviation, but does not perform well for medians, quantiles, and other rank-based statistics. For these sorts of statistics, a jackknife implementation that removes more than one observation can overcome this problem. The reason for this failure becomes apparent upon examination of the upper panel of figure 4.4: for σ_G, the vast majority of jackknife samples yield one of three discrete values! Because quartiles are insensitive to the removal of outliers, all samples created by the removal of a point larger than q_{75} lead to precisely the same estimate. The same is true for removal of any point smaller than q_{25}, and for any point in the range $q_{25} < x < q_{75}$. Because of this, the jackknife cannot accurately sample the error distribution, which leads to a gross misestimate of the result.

Should one use bootstrap or jackknife in practice? Although based on different approximations, they typically produce similar results for smooth statistics, especially for large samples. Jackknife estimates are usually easier to calculate, easier to apply to complex sampling schemes, and they also automatically remove bias. However, bootstrap is better for computing confidence intervals because it does not involve the assumption of asymptotic normality (i.e., it maps out the shape of the distribution). Note that bootstrap gives slightly different results even if the data set is fixed (because random resampling is performed), while jackknife gives repeatable results for a given data set (because all possible permutations are used). Of course, when feasible, it is prudent to use both bootstrap and jackknife and critically compare their results. Both methods should be used with caution when N is small. Again, before applying bootstrap to complex problems, consult the specialist literature (a good entry point is [6]).

Cross-validation and bootstrap aggregating (bagging) are methods closely related to jackknife and bootstrap. They are used in regression and classification contexts, and are discussed in §8.11 and §9.7, respectively.

AstroML contains some routines for performing basic nonparametric jackknife and bootstrap: `astroML.resample.bootstrap` and `astroML.resample.jackknife`.

```
>>> import numpy as np
>>> from astroML.resample import jackknife
>>> x = np.random.normal(loc=0, scale=1, size=1000)
>>> jackknife(x, np.std, kwargs=dict(ddof=1, axis=1))
(1.01, 0.02)
```

The standard deviation is found to be 1.01 ± 0.02. For more examples of the use of bootstrap and jackknife methods, see the source code of figure 4.3.

4.6. Hypothesis Testing

A common problem in statistics is to ask whether a given sample is consistent with some hypothesis. For example, we might be interested in whether a measured value x_i, or the whole set $\{x_i\}$, is consistent with being drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. Here $\mathcal{N}(\mu, \sigma)$ is our *null hypothesis*, typically corresponding to a "no effect" case, and we are trying to *reject it* in order to demonstrate that we measured some effect. A good example from astronomy is the source detection in images with substantial background (e.g., atmospheric sky brightness in optical images). Because the background fluctuates, the contribution of the source flux to a particular image resolution element must be substantially larger than the background fluctuation to represent a robust detection. Here, the null hypothesis is that the measured brightness in a given resolution element is due to background, and when we can reject it, we have a source detection. It is always assumed that we know how to compute the probability of a given outcome from the null hypothesis: for example, given the cumulative distribution function, $0 \leq H_0(x) \leq 1$ (see eq. 1.1), the probability that we would get a value at least as large as x_i is $p(x > x_i) = 1 - H(x_i)$, and is called the *p value*. Typically, a threshold p value is adopted, called *the significance level* α, and the null hypothesis is rejected when $p \leq \alpha$ (e.g., if $\alpha = 0.05$ and $p < 0.05$, the null hypothesis is rejected at a 0.05 significance level). If we fail to reject a hypothesis, it does not mean that we proved its correctness because it may be that our sample is simply not large enough to detect an effect.

For example, if we flip a coin 10 times and get 8 tails, should we reject the hypothesis that the coin is fair? If it is indeed fair, the binomial distribution (eq. 3.50) predicts that the probability of 8 or more tails is 0.054 and thus we cannot reject the null hypothesis at the 0.05 significance level. We shall return to this coin-flip example when discussing Bayesian methods in chapter 5.

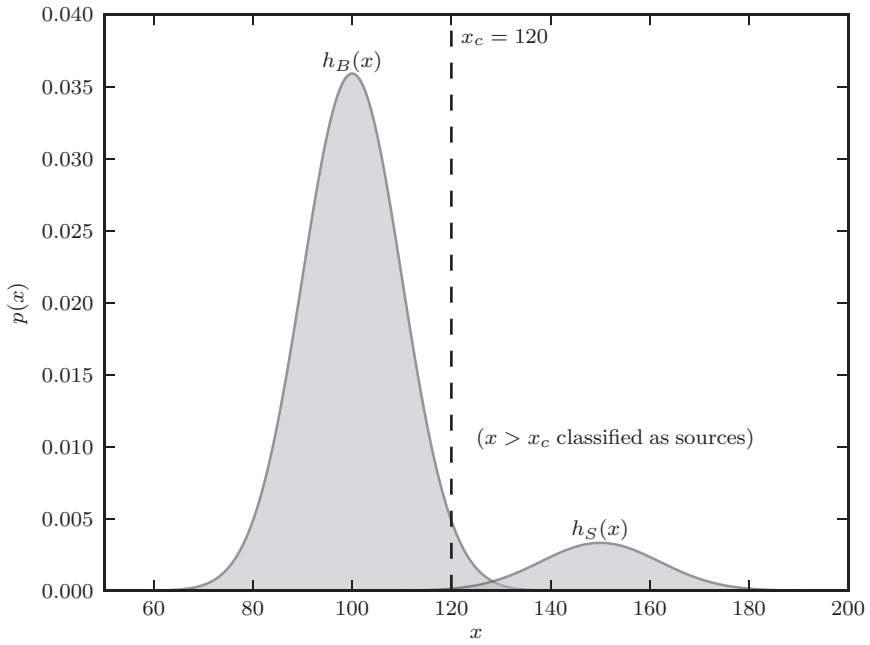

Figure 4.5. An example of a simple classification problem between two Gaussian distributions. Given a value of x, we need to assign that measurement to one of the two distributions (background vs. source). The cut at $x_c = 120$ leads to very few Type II errors (i.e., *false negatives*: points from the distribution h_S with $x < x_c$ being classified as background), but this comes at the cost of a significant number of Type I errors (i.e., *false positives*: points from the distribution h_B with $x > x_c$ being classified as sources).

When performing these tests, we are bound to make two types of errors, which statisticians memorably call *Type I and Type II errors* (Jerzy Neyman and Egon Pearson introduced this notation in 1933). Type I errors are cases when the null hypothesis is true but incorrectly rejected. In the context of source detection, these errors represent spurious sources, or more generally, false positives (see figure 4.5). The false-positive probability when testing a single datum is limited by the adopted significance level α (the case of multiple tests is discussed in the next section). Cases when the null hypothesis is false, but it is not rejected are called Type II errors (missed sources, or false negatives). The false-negative probability when testing a single datum is usually called β, and is related to *the power of a test* as $(1 - \beta)$. Hypothesis testing is intimately related to comparisons of distributions, as discussed below. The classical approach to hypothesis testing is not identical to the Bayesian approach, and we shall return to this topic in chapter 5 (see §5.4).

4.6.1. Simple Classification and Completeness vs. Contamination Trade-Off

As the significance level α is decreased (the criterion for rejecting the null hypothesis becomes more conservative), the number of false positives decreases and the number of false negatives increases. Therefore, there is a trade-off to be made to find an optimal value of α, which depends on the relative importance of false negatives and

positives in a particular problem. For example, if the null hypothesis is "my parachute is good," we are more concerned about false negatives (it's bad but we accept it as good) than about false positives (it's good but we reject it as bad) because the former can kill us while the latter presumably has less dire consequences (what α would make you feel safe in this case?). On the other hand, if the null hypothesis is "this undergraduate student would do great in graduate school," then accepting a bad student (false positive) is arguably less harmful than rejecting a truly good student (false negative).

When many instances of hypothesis testing are performed, a process called *multiple hypothesis testing*, the fraction of false positives can significantly exceed the value of α. The fraction of false positives depends not only on α and the number of data points, but also on the number of true positives (the latter is proportional to the number of instances when an alternative hypothesis is true). We shall illustrate these trade-offs with an example.

Often the underlying distribution from which data $\{x_i\}$ were drawn, $h(x)$, is a sum of two populations

$$h(x) = (1 - a)\, h_B(x) + a\, h_S(x), \tag{4.36}$$

where a is the relative normalization factor (we assume that integrals of h_B and h_S are normalized to unity). In this example there is not only a null hypothesis (B, for background), but also a specific alternative hypothesis (S, for source). Given $\{x_i\}$, for example counts obtained with a measuring apparatus, we want to assign to each individual measurement x_i the probability that it belongs to population S, $p_S(x_i)$ (of course, $p_B(x_i) = 1 - p_S(x_i)$ as there are only these two possibilities). Recall that the size of sample $\{x_i\}$ is N, and thus the number of *true* sources in the sample is Na. A simplified version of this problem is *classification*, where we assign the class S or B without retaining the knowledge of the actual probability p_S. In order for classification based on x to be possible at all, obviously $h_B(x)$ and $h_S(x)$ must be different (how to "measure" the difference between two distributions is discussed in §4.7).

If we choose a classification boundary value x_c, then the *expected number* of spurious sources (false positives or Type I errors) in the classified sample is

$$n_{\text{spurious}} = N\,(1 - a)\,\alpha = N\,(1 - a) \int_{x_c}^{\infty} h_B(x)\, dx, \tag{4.37}$$

and the number of missed sources (false negatives or Type II errors) is

$$n_{\text{missed}} = N\, a\, \beta = N\, a \int_{0}^{x_c} h_S(x)\, dx. \tag{4.38}$$

The number of instances classified as a source, that is, instances when the null hypothesis is rejected, is

$$n_{\text{source}} = N\, a\, - n_{\text{missed}} + n_{\text{spurious}}. \tag{4.39}$$

The sample *completeness* (also called sensitivity and recall rate in the statistics literature) is defined as

$$\eta = \frac{Na - n_{\rm missed}}{Na} = 1 - \int_0^{x_c} h_S(x)\, dx, \tag{4.40}$$

with $0 \leq \eta \leq 1$, and the sample *contamination* is defined as

$$\epsilon = \frac{n_{\rm spurious}}{n_{\rm source}}, \tag{4.41}$$

with $0 \leq \epsilon \leq 1$ (the $1 - \epsilon$ rate is sometimes called classification efficiency). The sample contamination is also called the *false discovery rate* (FDR). As x_c increases, the sample contamination decreases (good), but at the same time completeness decreases too (bad). This trade-off can be analyzed using the so-called *receiver operating characteristic* (ROC) curve which typically plots the fraction of true positives vs. the fraction of true negatives (see HTF09). In astronomy, ROC curves are often plotted as expected completeness vs. contamination (or sometimes efficiency) rate. The position along the ROC curve is parametrized by x_c (i.e., by the classification rule). The area under the ROC curve, sometimes called the c statistic, can be used to quantify overall performance of the classification method (see §9.8).

Note that the sample completeness involves neither N nor a, but the sample contamination depends on a: obviously, for $a = 1$ we get $\epsilon = 0$ (there can be no contamination if only sources exist), but more concerningly, $\lim_{a\to 0}(\epsilon)$=1. In other words, for small a, that is, when the true fraction of instances with a false null hypothesis is small, we can have a large sample contamination even when x_c corresponds to a very small value of α. How do we choose an optimal value of x_c?

To have a concrete example, we will use the source detection example and assume that $h_B(x) = \mathcal{N}(\mu = 100, \sigma = 10)$ and $h_S(x) = \mathcal{N}(\mu = 150, \sigma = 12)$, with $a = 0.1$ and $N = 10^6$ (say, an image with 1000 by 1000 resolution elements; the x values correspond to the sum of background and source counts). For illustration, see figure 4.5. If we naively choose $x_c = 120$ (a "2σ cut" away from the mean for h_B, corresponding to a Type I error probability of $\alpha = 0.024$), 21,600 values will be incorrectly classified as a source. With $x_c = 120$, the sample completeness is 0.994 and 99,400 values are correctly classified as a source. Although the Type I error rate is only 0.024, the sample contamination is $21{,}600/(21{,}600 + 99{,}400) = 0.179$, or over 7 times higher! Of course, this result that $\epsilon \gg \alpha$ is a consequence of the fact that the true population contains 9 times as many background values as it contains sources ($a = 0.1$).

In order to decrease the expected contamination level ϵ, we need to increase x_c, but the optimal value depends on a. Since a is often unknown in practice, choosing x_c is not straightforward. A simple practical method for choosing the optimal value of x_c for a given desired ϵ (or FDR) was proposed by Benjamini and Hochberg [2].

The Benjamini and Hochberg method assumes that measurements can be described by eq. 4.36 and makes an additional assumption that $h_B(x)$ is known (e.g., when a is small, it is possible to isolate a portion of an image to measure background count distribution). Given $h_B(x)$, and its cumulative counterpart $H_B(x)$, it is possible to assign a p value to each value in $\{x_i\}$ as $p_i = 1 - H_B(x_i)$, and sort the sample

so that p_i are increasing. If all $\{x_i\}$ values were drawn from $h_B(x)$, the differential distribution of these p_i values would be a uniform distribution by construction, and its cumulative distribution, $1 \leq C_i \leq N$, would increase linearly as

$$C_i^B = N\, p_i. \tag{4.42}$$

Instead, for Na cases in the adopted model the null hypothesis is false and they will result in an excess of small p_i values; hence, the observed cumulative distribution, $C_i = C(p_i) = i$, will have values much larger than C_i^B for small p. Benjamini and Hochberg realized that this fact can be used to find a classification threshold p_c (and the corresponding x_c and its index $i_c = C(p_c)$; recall that the sample is sorted by p_i) that guarantees that the sample contamination ϵ is below some desired value ϵ_0. Their proposal for finding p_c is very elegant and does not require involved computations: assume that the null hypothesis is rejected for all values $p_i \leq p_c$, resulting in a subsample of $i_c = C(p_c)$ values (i.e., these i_c values are selected as sources). The number of cases when the null hypothesis was actually true and falsely rejected is $(1-a)Np_c < Np_c$, and thus the contamination rate is

$$\epsilon = \frac{(1-a)Np_c}{i_c} < \frac{Np_c}{i_c}. \tag{4.43}$$

Therefore, the threshold value must satisfy

$$i_c < N\frac{p_c}{\epsilon_0}. \tag{4.44}$$

This condition corresponds to the intersection of the measured $C_i(p_i)$ curve and the straight line $C = Np/\epsilon_0$, and the algorithm is simply to find the largest value of i $(= C_i)$ which satisfies eq. 4.44 (see figure 4.6). Note that $p_c < \epsilon_0$ because $i_c < N$; if one naively adopted $p_c = \epsilon_0$, the resulting expected sample contamination would be a factor of N/i_c larger than ϵ_0.

The Benjamini and Hochberg algorithm is conservative because it assumes that $(1-a) \approx 1$ when deriving the upper bound on ϵ. Hence, the resulting contamination rate ϵ is a factor of $(1-a)$ smaller than the maximum allowed value ϵ_0. If we knew a, we could increase i_c given by eq. 4.44 by a factor of $1/(1-a)$, and thus increase the sample completeness, while still guaranteeing that the sample contamination does not exceed ϵ_0.

In cases where one can assume that large values of p (say, for $p > 0.5$) are dominated by the null hypothesis, which is often the case, the cumulative distribution is

$$C(p_i) = Na + N(1-a)\, p_i \quad \text{for } p_i > 0.5, \tag{4.45}$$

with, say, $C(p_i = 0.5) = C_{0.5}$. Given that the slope of this line is $2(N - C_{0.5})$, the number of cases when the null hypothesis is true but falsely rejected can be estimated as $2(N-C_{0.5})\,p_c$. This estimate amounts to scaling $h_B(x)$ to fit the observed

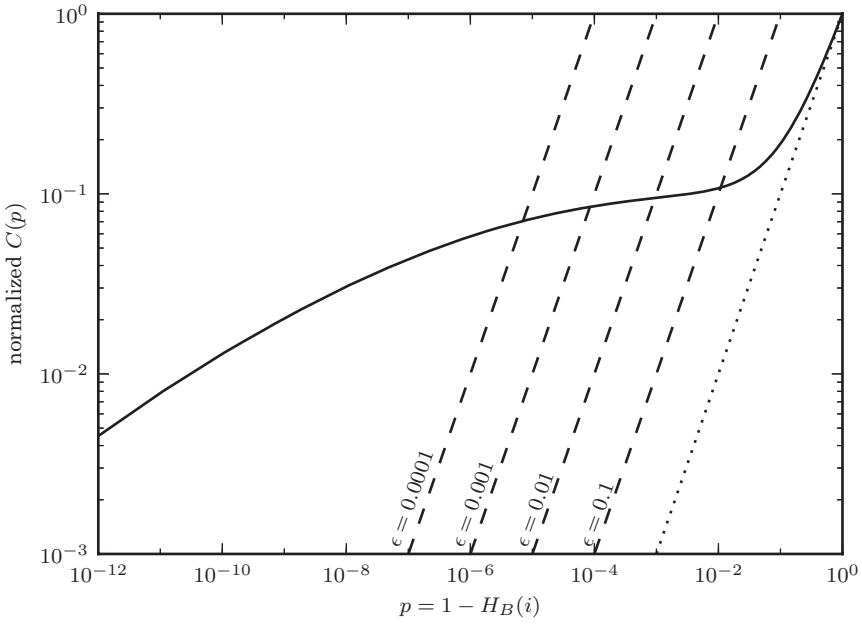

Figure 4.6. Illustration of the Benjamini and Hochberg method for 10^6 points drawn from the distribution shown in figure 4.5. The solid line shows the cumulative distribution of observed p values, normalized by the sample size. The dashed lines show the cutoff for various limits on contamination rate ϵ computed using eq. 4.44 (the accepted measurements are those with p smaller than that corresponding to the intersection of solid and dashed curves). The dotted line shows how the distribution would look in the absence of sources. The value of the cumulative distribution at $p = 0.5$ is 0.55, and yields a correction factor $\lambda = 1.11$ (see eq. 4.46).

distribution, or equivalently, estimating $(1 - a)$ as

$$\lambda^{-1} \equiv 1 - a = 2\left(1 - \frac{C_{0.5}}{N}\right). \tag{4.46}$$

Thus, the Benjamini and Hochberg method can be improved by multiplying i_c by λ, yielding the sample completeness increased by a factor λ.

4.7. Comparison of Distributions

We often ask whether two samples are drawn from the same distribution, or equivalently whether two sets of measurements imply a difference in the measured quantity. A similar question is whether a sample is consistent with being drawn from some known distribution (while real samples are always finite, the second question is the same as the first one when one of the samples is considered as infinitely large). In general, obtaining answers to these questions can be very complicated. First, what do we mean by "the same distribution"? Distributions can be described by their location, scale, and shape. When the distribution shape is assumed known, for example when we know for one or another reason that the sample is drawn

from a Gaussian distribution, the problem is greatly simplified to the consideration of only two parameters (location and scale, μ and σ from $\mathcal{N}(\mu, \sigma)$). Second, we might be interested in only one of these two parameters; for example, do two sets of measurements with different measurement errors imply the same mean value (e.g., two experimental groups measure the mass of the same elementary particle, or the same planet, using different methods).

Depending on data type (discrete vs. continuous random variables) and what we can assume (or not) about the underlying distributions, and the specific question we ask, we can use different statistical tests. The underlying idea of statistical tests is to use data to compute an appropriate statistic, and then compare the resulting data-based value to its expected distribution. The expected distribution is evaluated by *assuming that the null hypothesis is true*, as discussed in the preceding section. When this expected distribution implies that the data-based value is unlikely to have arisen from it by chance (i.e., the corresponding p value is small), the null hypothesis is rejected with some threshold probability α, typically 0.05 or 0.01 ($p < \alpha$). For example, if the null hypothesis is that our datum came from the $\mathcal{N}(0, 1)$ distribution, then $x = 3$ corresponds to $p = 0.003$ (see §3.3.2). Note again that $p > \alpha$ does *not* mean that the hypothesis is *proven* to be correct!

The number of various statistical tests in the literature is overwhelming and their applicability is often hard to discern. We describe here only a few of the most important tests, and further discuss hypothesis testing and distribution comparison in the Bayesian context in chapter 5.

4.7.1. Regression toward the Mean

Before proceeding with statistical tests for comparing distributions, we point out a simple statistical selection effect that is sometimes ignored and leads to spurious conclusions.

If two instances of a data set $\{x_i\}$ are drawn from some distribution, the mean difference between the matched values (i.e., the ith value from the first set and the ith value from the second set) will be zero. However, if we use one data set to select a subsample for comparison, the mean difference may become biased. For example, if we subselect the lowest quartile from the first data set, then the mean difference between the second and the first data set will be larger than zero.

Although this subselection step may sound like a contrived procedure, there are documented cases where the impact of a procedure designed to improve students' test scores was judged by applying it only to the worst performing students. Given that there is always some randomness (measurement error) in testing scores, these preselected students would have improved their scores without any intervention. This effect is called "regression toward the mean": if a random variable is extreme on its first measurement, it will tend to be closer to the population mean on a second measurement. In an astronomical context, a common related tale states that weather conditions observed at a telescope site today are, typically, not as good as those that would have been inferred from the prior measurements made during the site selection process.

Therefore, when selecting a subsample for further study, or a control sample for comparison analysis, one has to worry about various statistical selection effects. Going back to the above example with student test scores, a proper assessment of

a new educational procedure should be based on a randomly selected subsample of students who will undertake it.

4.7.2. Nonparametric Methods for Comparing Distributions

When the distributions are not known, tests are called nonparametric, or distribution-free tests. The most popular nonparametric test is the Kolmogorov–Smirnov (K-S) test, which compares the cumulative distribution function, $F(x)$, for two samples, $\{x1_i\}$, $i = 1, \ldots, N_1$ and $\{x2_i\}$, $i = 1, \ldots, N_2$ (see eq. 1.1 for definitions; we sort the sample and divide the rank (recall §3.6.1) of x_i by the sample size to get $F(x_i)$; $F(x)$ is a step function that increases by $1/N$ at each data point; note that $0 \leq F(x) \leq 1$).

The K-S test and its variations can be performed in Python using the routines `kstest`, `ks_2samp`, and `ksone` from the module `scipy.stats`:

```
>>> import numpy as np
>>> from scipy import stats
>>> vals = np.random.normal(loc=0, scale=1,
                            size=1000)
>>> stats.kstest(vals, "norm")
(0.0255, 0.529)
```

The D value is 0.0255, and the p value is 0.529. For more examples of these statistics, see the SciPy documentation, and the source code for figure 4.7.

The K-S test is based on the following statistic which measures the maximum distance of the two cumulative distributions $F_1(x1)$ and $F_2(x2)$:

$$D = \max |F_1(x1) - F_2(x2)| \tag{4.47}$$

($0 \leq D \leq 1$; we note that other statistics could be used to measure the difference between F_1 and F_2, e.g., the integrated square error). The key question is how often would the value of D computed from the data arise by chance if the two samples were drawn from the *same* distribution (the null hypothesis in this case). Surprisingly, this question has a well-defined answer even when we know nothing about the underlying distribution. Kolmogorov showed in 1933 (and Smirnov published tables with the numerical results in 1948) that the probability of obtaining by chance a value of D larger than the measured value is given by the function

$$Q_{\mathrm{KS}}(\lambda) = 2 \sum_{k=1}^{\infty} (-1)^{k-1} e^{-2k^2\lambda^2}, \tag{4.48}$$

where the argument λ can be accurately described by the following approximation (as shown by Stephens in 1970; see discussion in NumRec):

$$\lambda = \left(0.12 + \sqrt{n_e} + \frac{0.11}{\sqrt{n_e}}\right) D, \tag{4.49}$$

where the "effective" number of data points is computed from

$$n_e = \frac{N_1 N_2}{N_1 + N_2}. \tag{4.50}$$

Note that for large n_e, $\lambda \approx \sqrt{n_e} D$. If the probability that a given value of D is due to chance is very small (e.g., 0.01 or 0.05), we can reject the null hypothesis that the two samples were drawn from the same underlying distribution.

For n_e greater than about 10 or so, we can bypass eq. 4.48 and use the following simple approximation to evaluate D corresponding to a given probability α of obtaining a value at least that large:

$$D_{\mathrm{KS}} = \frac{C(\alpha)}{\sqrt{n_e}}, \tag{4.51}$$

where $C(\alpha)$ is the critical value of the Kolmogorov distribution with $C(\alpha = 0.05) = 1.36$ and $C(\alpha = 0.01) = 1.63$. Note that the ability to reject the null hypothesis (if it is really false) increases with $\sqrt{n_e}$. For example, if $n_e = 100$, then $D > D_{\mathrm{KS}} = 0.163$ would arise by chance in only 1% of all trials. If the actual data-based value is indeed 0.163, we can reject the null hypothesis that the data were drawn from the same (unknown) distribution, with our decision being correct in 99 out of 100 cases.

We can also use the K-S test to ask, "Is the measured $f(x)$ consistent with a known reference distribution function $h(x)$?" (When $h(x)$ is a Gaussian distribution with known parameters, it is more efficient to use the parametric tests described in the next section.) This case is called the "one-sample" K-S test, as opposed to the "two-sample" K-S test discussed above. In this case, $N_1 = N$ and $N_2 = \infty$, and thus $n_e = N$. Again, a small value of Q_{KS} (or $D > D_{\mathrm{KS}}$) indicates that it is unlikely, at the given confidence level set by α, that the data summarized by $f(x)$ were drawn from $h(x)$.

The K-S test is sensitive to the location, the scale, and the shape of the underlying distribution(s) and, because it is based on cumulative distributions, it is invariant to reparametrization of x (we would obtain the same conclusion if, for example, we used $\ln x$ instead of x). The main strength but also the main weakness of the K-S test is its ignorance about the underlying distribution. For example, the test is insensitive to details in the differential distribution function (e.g., narrow regions where it drops to zero), and more sensitive near the center of the distribution than at the tails (the K-S test is not the best choice for distinguishing samples drawn from Gaussian and exponential distributions; see §4.7.4).

For an example of the two-sample K-S test, refer to figure 3.25, where it is used to confirm that two random samples are drawn from the same underlying data set. For an example of the one-sample K-S test, refer to figure 4.7, where it is compared to other tests of Gaussianity.

A simple test related to the K-S test was developed by Kuiper to treat distributions defined on a circle. It is based on the statistic

$$D^* = \max\{F_1(x1) - F_2(x2)\} + \max\{F_2(x1) - F_1(x2)\}. \tag{4.52}$$

As is evident, this statistic considers both positive and negative differences between two distributions (D from the K-S test is equal to the greater of the two terms).

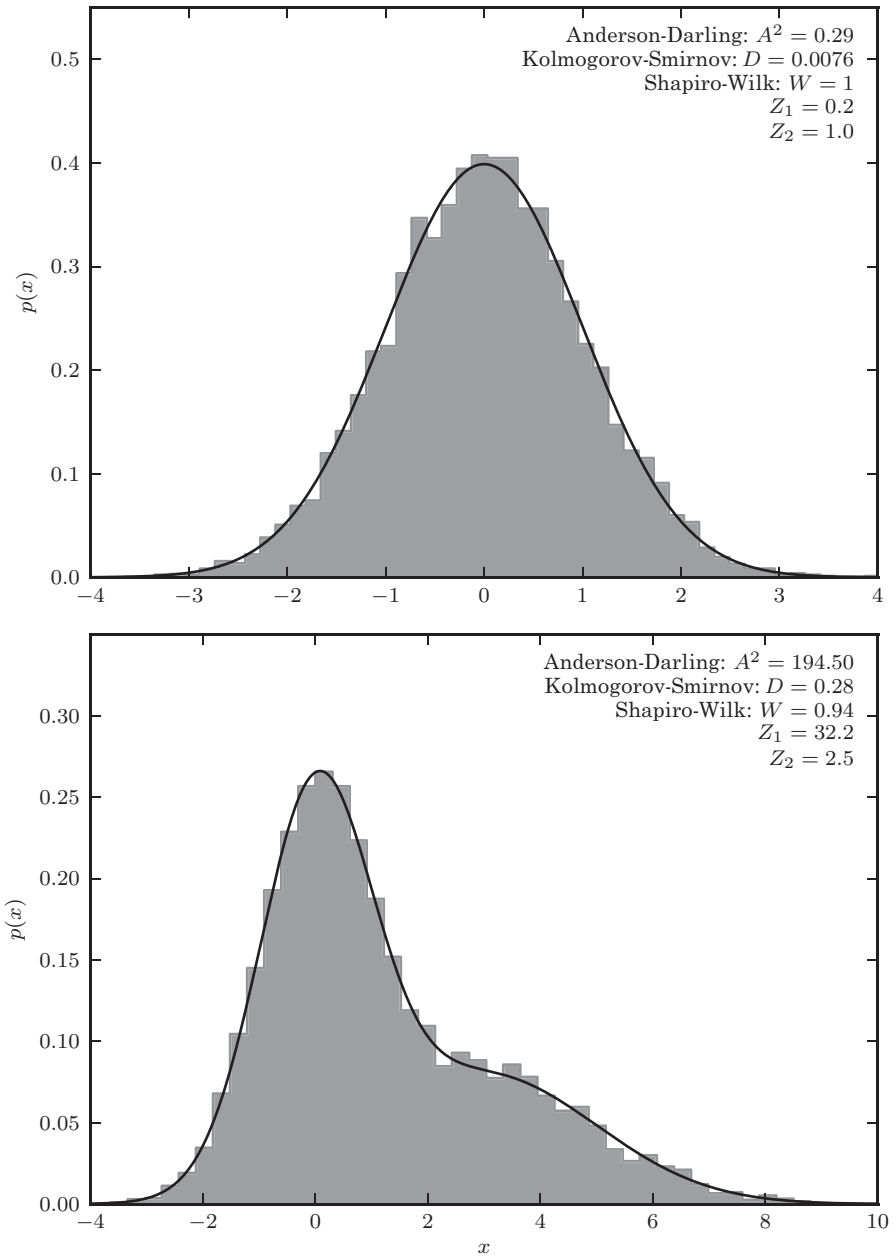

Figure 4.7. The results of the Anderson–Darling test, the Kolmogorov–Smirnov test, and the Shapiro–Wilk test when applied to a sample of 10,000 values drawn from a normal distribution (upper panel) and from a combination of two Gaussian distributions (lower panel).

For distributions defined on a circle (i.e., $0^\circ < x < 360^\circ$), the value of D^* is invariant to where exactly the origin ($x = 0^\circ$) is placed. Hence, the Kuiper test is a good test for comparing the longitude distributions of two astronomical samples. By analogy

with the K-S test,

$$Q_{\rm Kuiper}(\lambda) = 2\sum_{k=1}^{\infty}(4k^2\lambda^2 - 1)\, e^{-2k^2\lambda^2}, \tag{4.53}$$

with

$$\lambda = \left(0.155 + \sqrt{n_e} + \frac{0.24}{\sqrt{n_e}}\right) D^*. \tag{4.54}$$

The K-S test is not the only option for nonparametric comparison of distributions. The Cramér–von Mises criterion, the Watson test, and the Anderson–Darling test, to name but a few, are similar in spirit to the K-S test, but consider somewhat different statistics. For example, the Anderson–Darling test is more sensitive to differences in the tails of the two distributions than the K-S test. A practical difficulty with these other statistics is that a simple summary of their behavior, such as given by eq. 4.48 for the K-S test, is not readily available. We discuss a very simple test for detecting non-Gaussian behavior in the tails of a distribution in §4.7.4.

A somewhat similar quantity that is also based on the cumulative distribution function is the Gini coefficient (developed by Corrado Gini in 1912). It measures the deviation of a given cumulative distribution ($F(x)$, defined for $x_{\rm min} \leq x \leq x_{\rm max}$) from that expected for a uniform distribution:

$$G = 1 - 2\int_{x_{\rm min}}^{x_{\rm max}} F(x)\, dx. \tag{4.55}$$

When $F(x)$ corresponds to a uniform differential distribution, $G = 0$, and $G \leq 1$ always. The Gini coefficient is *not* a statistical test, but we mention it here for reference because it is commonly used in classification (see §9.7.1), in economics and related fields (usually to quantify income inequality), and sometimes confused with a statistical test.

The *U* test and the Wilcoxon test

The U test and Wilcoxon test are implemented in `mannwhitneyu` and `ranksums` (i.e., Wilcoxon rank-sum test) within the `scipy.stats` module:

```
>>> import numpy as np
>>> from scipy import stats
>>> x, y = np.random.normal(0, 1, size=(2, 1000))
>>> stats.mannwhitneyu(x, y)
(487678.0, 0.1699)
```

The U test result is close to the expected $N_1 N_2/2$, indicating that the two samples are drawn from the same distribution. For more information, see the SciPy documentation.

Nonparametric methods for comparing distributions, for example, the K-S test, are often sensitive to more than a single distribution property, such as the location or

scale parameters. Often, we are interested in differences in only a particular statistic, such as the mean value, and do not care about others. There are several widely used nonparametric tests for such cases. They are analogous to the better-known classical parametric tests, the *t* test and the paired *t* test (which assume Gaussian distributions and are described below), and are based on the ranks of data points, rather than on their values.

The *U* test, or the Mann–Whitney–Wilcoxon test (or the Wilcoxon rank-sum test, not to be confused with the Wilcoxon signed-rank test described below) is a nonparametric test for testing whether two data sets are drawn from distributions with different location parameters (if these distributions are known to be Gaussian, the standard classical test is called the *t* test, described in §4.7.6). The sensitivity of the *U* test is dominated by a difference in medians of the two tested distributions.

The *U* statistic is determined using the ranks for the full sample obtained by concatenating the two data sets and sorting them, while retaining the information about which data set a value came from. To compute the *U* statistic, take each value from sample 1 and count the number of observations in sample 2 that have a smaller rank (in the case of identical values, take half a count). The sum of these counts is *U*, and the minimum of the values with the samples reversed is used to assess the significance. For cases with more than about 20 points per sample, the *U* statistic for sample 1 can be more easily computed as

$$U_1 = R_1 - \frac{N_1(N_1 - 1)}{2}, \tag{4.56}$$

where R_1 is the sum of ranks for sample 1, and analogously for sample 2. The adopted *U* statistic is the smaller of the two (note that $U_1 + U_2 = N_1 N_2$, which can be used to check computations). The behavior of *U* for large samples can be well approximated with a Gaussian distribution, $\mathcal{N}(\mu_U, \sigma_U)$, of variable

$$z = \frac{U - \mu_U}{\sigma_U}, \tag{4.57}$$

with

$$\mu_U = \frac{N_1\, N_2}{2} \tag{4.58}$$

and

$$\sigma_U = \sqrt{\frac{N_1\, N_2\, (N_1 + N_2 + 1)}{12}}. \tag{4.59}$$

For small data sets, consult the literature or use one of the numerous and widely available statistical programs.

A special case of comparing the means of two data sets is when the data sets have the same size ($N_1 = N_2 = N$) and data points are paired. For example, the two data sets could correspond to the same sample measured twice, "before" and "after" something that could have affected the values, and we are testing for evidence of a change in mean values. The nonparametric test that can be used to compare means of two arbitrary distributions is the Wilcoxon signed-rank test. The test is based on

differences $y_i = x1_i - x2_i$, and the values with $y_i = 0$ are excluded, yielding the new sample size $m \leq N$. The sample is ordered by $|y_i|$, resulting in the rank R_i for each pair, and each pair is assigned $\Phi_i = 1$ if $x1_i > x2_i$ and 0 otherwise. The Wilcoxon signed-ranked statistic is then

$$W_+ = \sum_i^m \Phi_i R_i, \tag{4.60}$$

that is, all the ranks with $y_i > 0$ are summed. Analogously, W_- is the sum of all the ranks with $y_i < 0$, and the statistic T is the smaller of the two. For small values of m, the significance of T can be found in tables. For m larger than about 20, the behavior of T can be well approximated with a Gaussian distribution, $\mathcal{N}(\mu_T, \sigma_T)$, of the variable

$$z = \frac{T - \mu_T}{\sigma_T}, \tag{4.61}$$

with

$$\mu_T = \frac{N(2N+1)}{2} \tag{4.62}$$

and

$$\sigma_T = N\sqrt{\frac{(2N+1)}{12}}. \tag{4.63}$$

The Wilcoxon signed-rank test can be performed with the function `scipy.stats.wilcoxon`:

```
import numpy as np
from scipy import stats
x, y = np.random.normal(0, 1, size=(2, 1000))
T, p = stats.wilcoxon(x, y)
```

See the documentation of the `wilcoxon` function for more details.

4.7.3. Comparison of Two-Dimensional Distributions

There is no direct analog of the K-S test for multidimensional distributions because cumulative probability distribution is not well defined in more than one dimension. Nevertheless, it is possible to use a method similar to the K-S test, though not as straightforward (developed by Peacock in 1983, and Fasano and Franceschini in 1987; see §14.7 in NumRec), as follows.

Given two sets of points, $\{x_i^A, y_i^A\}$, $i = 1, \ldots, N_A$ and $\{x_i^B, y_i^B\}$, $i = 1, \ldots, N_B$, define four quadrants centered on the point (x_j^A, y_j^A) and compute the fraction of data points from each data set in each quadrant. Record the maximum difference (among the four quadrants) between the fractions for data sets A and B. Repeat for all points from sample A to get the overall maximum difference, D_A, and repeat the whole procedure for sample B. The final statistic is then $D = (D_A + D_B)/2$.

Although it is not strictly true that the distribution of D is independent of the details of the underlying distributions, Fasano and Franceschini showed that its variation is captured well by the coefficient of correlation, ρ (see eq. 3.81). Using simulated samples, they derived the following behavior (analogous to eq. 4.49 from the one-dimensional K-S test):

$$\lambda = \frac{\sqrt{n_e}\, D}{1 + (0.25 - 0.75/\sqrt{n_e})\sqrt{1 - \rho^2}}. \tag{4.64}$$

This value of λ can be used with eq. 4.48 to compute the significance level of D when $n_e > 20$.

4.7.4. Is My Distribution Really Gaussian?

When asking, "Is the measured $f(x)$ consistent with a known reference distribution function $h(x)$?", a few standard statistical tests can be used when we know, or can assume, that both $h(x)$ and $f(x)$ are Gaussian distributions. These tests are at least as efficient as any nonparametric test, and thus are the preferred option. Of course, in order to use them reliably we need to first convince ourselves (and others!) that our $f(x)$ is consistent with being a Gaussian.

Given a data set $\{x_i\}$, we would like to know whether we can reject the null hypothesis (see §4.6) that $\{x_i\}$ was drawn from a Gaussian distribution. Here we are not asking for specific values of the location and scale parameters, but only whether the *shape* of the distribution is Gaussian. In general, deviations from a Gaussian distribution could be due to nonzero skewness, nonzero kurtosis (i.e., thicker symmetric or asymmetric tails), or more complex combinations of such deviations. Numerous tests are available in statistical literature which have varying sensitivity to different deviations. For example, the difference between the mean and the median for a given data set is sensitive to nonzero skewness, but has no sensitivity whatsoever to changes in kurtosis. Therefore, if one is trying to detect a difference between the Gaussian $\mathcal{N}(\mu = 4, \sigma = 2)$ and the Poisson distribution with $\mu = 4$, the difference between the mean and the median might be a good test (0 vs. 1/6 for large samples), but it will not catch the difference between a Gaussian and an exponential distribution no matter what the size of the sample.

As already discussed in §4.6, a common feature of most tests is to predict the distribution of their chosen statistic under the assumption that the null hypothesis is true. An added complexity is whether the test uses any parameter estimates derived from data. Given the large number of tests, we limit our discussion here to only a few of them, and refer the reader to the voluminous literature on statistical tests in case a particular problem does not lend itself to these tests.

The first test is the Anderson–Darling test, specialized to the case of a Gaussian distribution. The test is based on the statistic

$$A^2 = -N - \frac{1}{N}\sum_{i=1}^{N}[(2i - 1)\ln(F_i) + (2N - 2i + 1)\ln(1 - F_i)], \tag{4.65}$$

TABLE 4.1.
The values of the Anderson–Darling statistic A^2 corresponding to significance level p.

μ and σ from data?	$p = 0.05$	$p = 0.01$
μ no, σ no	2.49	3.86
μ yes, σ no	1.11	1.57
μ no, σ yes	2.32	3.69
μ yes, σ yes	0.79	1.09

where F_i is the ith value of the cumulative distribution function of z_i, which is defined as

$$z_i = \frac{x_i - \mu}{\sigma}, \tag{4.66}$$

and assumed to be in ascending order. In this expression, either one or both of μ and σ can be known, or determined from data $\{x_i\}$. Depending on which parameters are determined from data, the statistical behavior of A^2 varies. Furthermore, if *both* μ and σ are determined from data (using eqs. 3.31 and 3.32), then A^2 needs to be multiplied by $(1 + 4/N - 25/N^2)$. The specialization to a Gaussian distribution enters when predicting the detailed statistical behavior of A^2, and its values for a few common significance levels (p) are listed in table 4.1. The values corresponding to other significance levels, as well as the statistical behavior of A^2 in the case of distributions other than Gaussian can be computed with simple numerical simulations (see the example below).

`scipy.stats.anderson` implements the Anderson–Darling test:

```
>>> import numpy as np
>>> from scipy import stats
>>> x = np.random.normal(0, 1, size=1000)
>>> A, crit, sig = stats.anderson(x, 'norm')
>>> A
0.54728
```

See the source code of figure 4.7 for a more detailed example.

Of course, the K-S test can also be used to detect a difference between $f(x)$ and $\mathcal{N}(\mu, \sigma)$. A difficulty arises if μ and σ are determined from the same data set: in this case the behavior of Q_{KS} is different from that given by eq. 4.48 and has only been determined using Monte Carlo simulations (and is known as the Lilliefors distribution [16]).

The third common test for detecting non-Gaussianity in $\{x_i\}$ is the Shapiro–Wilk test. It is implemented in a number of statistical programs, and details about this test can be found in [23]. Its statistic is based on both data values, x_i, and data

ranks, R_i (see §3.6.1):

$$W = \frac{\left(\sum_{i=1}^{N} a_i R_i\right)^2}{\sum_{i=1}^{N}(x_i - \overline{x})^2}, \tag{4.67}$$

where constants a_i encode the expected values of the order statistics for random variables sampled from the standard normal distribution (the test's null hypothesis). The Shapiro–Wilk test is very sensitive to non-Gaussian tails of the distribution ("outliers"), but not as much to detailed departures from Gaussianity in the distribution's core. Tables summarizing the statistical behavior of the W statistic can be found in [11].

The Shapiro–Wilk test is implemented in `scipy.stats.shapiro`:

```
>>> import numpy as np
>>> from scipy import stats
>>> x = np.random.normal(0, 1, 1000)
>>> stats.shapiro(x)
(0.9975, 0.1495)
```

A value of W close to 1 indicates that the data is indeed Gaussian. For more information, see the documentation of the function `shapiro`.

Often the main departure from Gaussianity is due to so-called "catastrophic outliers," or largely discrepant values many σ away from μ. For example, the overwhelming majority of measurements of fluxes of objects in an astronomical image may follow a Gaussian distribution, but, for just a few of them, unrecognized cosmic rays could have had a major impact on flux extraction. A simple method to detect the presence of such outliers is to compare the sample standard deviation s (eq. 3.32) and σ_G (eq. 3.36). Even when the outlier fraction is tiny, the ratio s/σ_G can become significantly large. When $N > 100$, for a Gaussian distribution (i.e., for the null hypothesis), this ratio follows a nearly Gaussian distribution with $\mu \sim 1$ and with $\sigma \sim 0.92/\sqrt{N}$. For example, if you measure $s/\sigma_G = 1.3$ using a sample with $N = 100$, then you can state that the probability of such a large value appearing by chance is less than 1%, and reject the null hypothesis that your sample was drawn from a Gaussian distribution. Another useful result is that the difference of the mean and the median drawn from a Gaussian distribution also follows a nearly Gaussian distribution with $\mu \sim 0$ and $\sigma \sim 0.76s/\sqrt{N}$. Therefore, when $N > 100$ we can define two simple statistics based on the measured values of (μ, q_{50}, s, and σ_G) that both measure departures in terms of Gaussian-like "sigma":

$$Z_1 = 1.3\,\frac{|\mu - q_{50}|}{s}\,\sqrt{N} \tag{4.68}$$

and

$$Z_2 = 1.1 \left| \frac{s}{\sigma_G} - 1 \right| \sqrt{N}. \tag{4.69}$$

Of course, these and similar results for the statistical behavior of various statistics can be easily derived using Monte Carlo samples (see §3.7).

Figure 4.7 shows the results of these tests when applied to samples of $N = 10{,}000$ values selected from a Gaussian distribution and from a mixture of two Gaussian distributions. To summarize, for data that depart from a Gaussian distribution, we expect the Anderson–Darling A^2 statistic to be much larger than 1 (see table 4.1), the K-S D statistic (see eq. 4.47 and 4.51) to be much larger than $1/\sqrt{N}$, the Shapiro–Wilk W statistic to be smaller than 1, and Z_1 and Z_2 to be larger than several σ. All these tests correctly identify the first data set as being normally distributed, and the second data set as departing from normality.

In cases when our empirical distribution fails the tests for Gaussianity, but there is no strong motivation for choosing an alternative specific distribution, a good approach for modeling non-Gaussianity is to adopt the Gram–Charlier series,

$$h(x) = \mathcal{N}(\mu, \sigma) \sum_{k=0}^{\infty} a_k \, H_k(z), \tag{4.70}$$

where $z = (x - \mu)/\sigma$, and $H_k(z)$ are the Hermite polynomials ($H_0 = 1$, $H_1 = z$, $H_2 = z^2 - 1$, $H_3 = z^3 - 3z$, etc.). For "nearly Gaussian" distributions, even the first few terms of the series provide a good description of $h(x)$ (see figure 3.6 for an example of using the Gram–Charlier series to generate a skewed distribution). A related expansion, the Edgeworth series, uses derivatives of $h(x)$ to derive "correction" factors for a Gaussian distribution.

4.7.5. Is My Distribution Bimodal?

It happens frequently in practice that we want to test a hypothesis that the data were drawn from a unimodal distribution (e.g., in the context of studying bimodal color distribution of galaxies, bimodal distribution of radio emission from quasars, or the kinematic structure of the Galaxy's halo). Answering this question can become quite involved and we discuss it in chapter 5 (see §5.7.3).

4.7.6. Parametric Methods for Comparing Distributions

Given a sample $\{x_i\}$ that does not fail any test for Gaussianity, one can use a few standard statistical tests for comparing means and variances. They are more efficient (they require smaller samples to reject the null hypothesis) than nonparametric tests, but often by much less than a factor of 2, and for good nonparametric tests close to 1 (e.g., the efficiency of the U test compared to the t test described below is as high as 0.95). Hence, nonparametric tests are generally the preferable option to classical tests which assume Gaussian distributions. Nevertheless, because of their ubiquitous presence in practice and literature, we briefly summarize the two most important classical tests. As before, we assume that we are given two samples, $\{x1_i\}$ with $i = 1, \ldots, N_1$, and $\{x2_i\}$ with $i = 1, \ldots, N_2$.

Comparison of Gaussian means using the *t* test

Variants of the t test can be computed using the routines `ttest_rel`, `ttest_ind`, and `ttest_1samp`, available in the module `scipy.stats`:

```
>>> import numpy as np
>>> from scipy import stats
>>> x, y = np.random.normal(size=(2, 1000))
>>> t, p = stats.ttest_ind(x, y)
```

See the documentation of the above SciPy functions for more details.

If the only question we are asking is whether our data $\{x1_i\}$ and $\{x2_i\}$ were drawn from two Gaussian distributions with a different μ but the same σ, and we were given σ, the answer would be simple. We would first compute the mean values for both samples, $\overline{x1}$ and $\overline{x2}$, using eq. 3.31, and their standard errors, $\sigma_{\overline{x1}} = \sigma/\sqrt{N_1}$ and analogously for $\sigma_{\overline{x2}}$, and then ask how large is the difference $\Delta = \overline{x1} - \overline{x2}$ in terms of its expected scatter, $\sigma_\Delta = \sigma\sqrt{1/N_1^2 + 1/N_2^2}$: $M_\sigma = \Delta/\sigma_\Delta$. The probability that the observed value of M would arise by chance is given by the Gauss error function (see §3.3.2) as $p = 1 - \mathrm{erf}(M/\sqrt{2})$. For example, for $M = 3$, $p = 0.003$.

If we do *not* know σ, but need to estimate it from data (with possibly different values for the two samples, s_1 and s_2; see eq. 3.32), then the ratio $M_s = \Delta/s_\Delta$, where $s_\Delta = \sqrt{s_1^2/N_1 + s_2^2/N_2}$, can no longer be described by a Gaussian distribution! Instead, it follows Student's t distribution (see the discussion in §5.6.1). The number of degrees of freedom depends on whether we assume that the two underlying distributions from which the samples were drawn have the same variances or not. If we can make this assumption then the relevant statistic (corresponding to M_s) is

$$t = \frac{\overline{x1} - \overline{x2}}{s_D}, \tag{4.71}$$

where

$$s_D = \sqrt{s_{12}^2 \left(\frac{1}{N_1} + \frac{1}{N_2}\right)} \tag{4.72}$$

is an estimate of the standard error of the difference of the means, and

$$s_{12} = \sqrt{\frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2}{N_1 + N_2 - 2}} \tag{4.73}$$

is an estimator of the common standard deviation of the two samples. The number of degrees of freedom is $k = (N_1 + N_2 - 2)$. Hence, instead of looking up the significance of $M_\sigma = \Delta/\sigma_\Delta$ using the Gaussian distribution $\mathcal{N}(0, 1)$, we use the significance corresponding to t and Student's t distribution with k degrees of freedom. For very large samples, this procedure tends to the simple case with known σ described in the

first paragraph because Student's t distribution tends to a Gaussian distribution (in other words, s converges to σ).

If we cannot assume that the two underlying distributions from which the samples were drawn have the same variances, then the appropriate test is called Welch's t test and the number of degrees of freedom is determined using the Welch–Satterthwaite equation (however, see §5.6.1 for the Bayesian approach). For formulas and implementation, see NumRec.

A special case of comparing the means of two data sets is when the data sets have the same size ($N_1 = N_2 = N$) and each pair of data points has the same σ, but the value of σ is not the same for all pairs (recall the difference between the nonparametric U and the Wilcoxon tests). In this case, the t test for paired samples should be used. The expression 4.71 is still valid, but eq. 4.72 needs to be modified as

$$s_D = \sqrt{\frac{(N_1 - 1)s_1^2 + (N_2 - 1)s_2^2 - 2\mathrm{Cov}_{12}}{N}}, \tag{4.74}$$

where the covariance between the two samples is

$$\mathrm{Cov}_{12} = \frac{1}{N-1}\sum_{i=1}^{N}(x1_i - \overline{x1})(x2_i - \overline{x2}). \tag{4.75}$$

Here the pairs of data points from the two samples need to be properly arranged when summing, and the number of degrees of freedom is $N - 1$.

Comparison of Gaussian variances using the F test

The F test can be computed using the routine `scipy.stats.f_oneway`:

```
>>> import numpy as np
>>> from scipy import stats
>>> x, y = np.random.normal(size=(2, 1000))
>>> F, p = stats.f_oneway(x, y)
```

See the SciPy documentation for more details.

The F test is used to compare the variances of two samples, $\{x1_i\}$ and $\{x2_i\}$, drawn from two unspecified Gaussian distributions. The null hypothesis is that the variances of two samples are equal, and the statistic is based on the ratio of the sample variances (see eq. 3.32),

$$F = \frac{s_1^2}{s_2^2}, \tag{4.76}$$

where F follows Fisher's F distribution with $d_1 = N_1 - 1$ and $d_2 = N_2 - 1$ (see §3.3.9). Situations when we are interested in only knowing whether $\sigma_1 < \sigma_2$ or $\sigma_2 < \sigma_1$ are treated by appropriately using the left and right tails of Fisher's F distribution.

We will conclude this section by quoting Wall and Jenkins: "The application of efficient statistical procedure has power; but the application of common sense has more." We will see in the next chapter that the Bayesian approach provides a transparent mathematical framework for quantifying our common sense.

4.8. Nonparametric Modeling and Histograms

When there is no strong motivation for adopting a parametrized description (typically an analytic function with free parameters) of a data set, nonparametric methods offer an alternative approach. Somewhat confusingly, "nonparametric" does not mean that there are no parameters. For example, one of the simplest nonparametric methods to analyze a one-dimensional data set is a histogram. To construct a histogram, we need to specify bin boundaries, and we implicitly assume that the estimated distribution function is piecewise constant within each bin. Therefore, here too there are parameters to be determined—the value of the distribution function in each bin. However, there is no specific distribution class, such as the set of all possible Gaussians, or Laplacians, but rather a general set of *distribution-free* models, called the Sobolev space. The Sobolev space includes all functions, $h(x)$, that satisfy some smoothness criteria, such as

$$\int [h''(x)]^2 dx < \infty. \tag{4.77}$$

This constraint, for example, excludes all functions with infinite spikes. Formally, a method is nonparametric if it provides a distribution function estimate $f(x)$ that approaches the true distribution $h(x)$ with enough data, for any $h(x)$ in a class of functions with relatively weak assumptions, such as the Sobolev space above.

Nonparametric methods play a central role in modern machine learning. They provide the highest possible predictive accuracies, as they can model any shape of distribution, down to the finest detail which still has predictive power, though they typically come at a higher computational cost than more traditional multivariate statistical methods. In addition, it is harder to interpret the results of nonparametric methods than those of parametric models.

Nonparametric methods are discussed extensively in the rest of this book, including methods such as nonparametric correction for the selection function in the context of luminosity function estimation (§4.9), kernel density estimation (§6.1.1), and decision trees (§9.7). In this chapter, we only briefly discuss one-dimensional histograms.

4.8.1. Histograms

A histogram can fit virtually any shape of distribution, given enough bins. This is the key—while each bin can be thought of as a simple constant estimator of the density in that bin, the overall histogram is a piecewise constant estimator which can be thought of as having a tuning parameter—the number of bins. When the number of data points is small, the number of bins should somehow be small, as there is not enough information to warrant many bins. As the number of data points grows, the

number of bins should also grow to capture the increasing amount of detail in the distribution's shape that having more data points allows. This is a general feature of nonparametric methods—they are composed of simple pieces, and the number of pieces grows with the number of data points.

Getting the number of bins right is clearly critical. Pragmatically, it can easily make the difference between concluding that a distribution has a single mode or that it has two modes. Intuitively, we expect that a large bin width will destroy fine-scale features in the data distribution, while a small width will result in increased counting noise per bin. We emphasize that it is *not* necessary to bin the data before estimating model parameters. A simple example is the case of data drawn from a Gaussian distribution. We can estimate its parameters μ and σ using eqs. 3.31 and 3.32 without ever binning the data. This is a general result that will be discussed in the context of arbitrary distributions in chapter 5. Nevertheless, binning can allow us to visualize our data and explore various features in order to motivate the model selection.

We will now look at a few rules of thumb for the surprisingly subtle question of choosing the critical bin width, based on frequentist analyses. The gold standard for frequentist bin width selection is cross-validation, which is more computationally intensive. This topic is discussed in §6.1.1, in the context of a generalization of histograms (kernel density estimation). However, because histograms are so useful as quick data visualization tools, simple rules of thumb are useful to have in order to avoid large or complex computations.

Various proposed methods for choosing optimal bin width typically suggest a value proportional to some estimate of the distribution's scale, and decreasing with the sample size. The most popular choice is "Scott's rule" which prescribes a bin width

$$\Delta_b = \frac{3.5\sigma}{N^{1/3}}, \tag{4.78}$$

where σ is the sample standard deviation, and N is the sample size. This rule asymptotically minimizes the mean integrated square error (see eq. 4.14) and assumes that the underlying distribution is Gaussian; see [22]. An attempt to generalize this rule to non-Gaussian distributions is the Freedman–Diaconis rule,

$$\Delta_b = \frac{2(q_{75} - q_{25})}{N^{1/3}} = \frac{2.7\sigma_G}{N^{1/3}}, \tag{4.79}$$

which estimates the scale ("spread") of the distribution from its interquartile range (see [12]). In the case of a Gaussian distribution, Scott's bin width is 30% larger than the Freedman–Diaconis bin width. Some rules use the extremes of observed values to estimate the scale of the distribution, which is clearly inferior to using the interquartile range when outliers are present.

Although the Freedman–Diaconis rule attempts to account for non-Gaussian distributions, it is too simple to distinguish, for example, multimodal and unimodal distributions that have the same σ_G. The main reason why finding the optimal bin size is not straightforward is that the result depends on both the actual data distribution and the choice of metric (such as the mean square error) to be optimized.

The interpretation of binned data essentially represents a model fit, where the model is a piecewise constant function. Different bin widths correspond to different models, and choosing the best bin width amounts to the selection of the best model. The model selection is a topic discussed in detail in chapter 5 on Bayesian statistical inference, and in that context we will describe a powerful method that is cognizant of the detailed properties of a given data distribution. We will also compare these three different rules using multimodal and unimodal distributions (see §5.7.2, in particular figure 5.20).

NumPy and Matplotlib contain powerful tools for creating histograms in one dimension or multiple dimensions. The Matplotlib command `pylab.hist` is the easiest way to plot a histogram:

```
In [1]: %pylab
In [2]: import numpy as np
In [3]: x = np.random.normal(size=1000)
In [4]: plt.hist(x, bins=50)
```

For more details, see the source code for the many figures in this chapter which show histograms. For computing but not plotting a histogram, the functions `numpy.histogram`, `numpy.histogram2d`, and `numpy.histogramdd` provide optimized implementations:

```
In [5]: counts, bins = np.histogram(x, bins=50)
```

The above rules of thumb for choosing bin widths are implemented in the submodule `astroML.density_estimation`, using the functions `knuth_bin_width`, `scotts_bin_width`, and `freedman_bin_width`. There is also a `pylab`-like interface for simple histogramming:

```
In [6]: from astroML.plotting import hist
In [7]: hist(x, bins='freedman') # can also choose
        # 'knuth' or 'scott'
```

The `hist` function in AstroML operates just like the `hist` function in Matplotlib, but can optionally use one of the above routines to choose the binning. For more details see the source code associated with figure 5.20, and the associated discussion in §5.7.2.

4.8.2. How to Determine the Histogram Errors?

Assuming that we have selected a bin size, Δ_b, the N values of x_i are sorted into M bins, with the count in each bin n_k, $k = 1, \ldots, M$. If we want to express the results as a properly normalized $f(x)$, with the values f_k in each bin, then it is customary to

adopt

$$f_k = \frac{n_k}{\Delta_b N}. \tag{4.80}$$

The unit for f_k is the inverse of the unit for x_i.

Each estimate of f_k comes with some uncertainty. It is customary to assign "error bars" for each n_k equal to $\sqrt{n_k}$ and thus the uncertainty of f_k is

$$\sigma_k = \frac{\sqrt{n_k}}{\Delta_b N}. \tag{4.81}$$

This practice assumes that n_k are scattered around the true values in each bin (μ) according to a Gaussian distribution, and that error bars enclose the 68% confidence range for the true value. However, when counts are low this assumption of Gaussianity breaks down and the Poisson distribution should be used instead. For example, according to the Gaussian distribution, negative values of μ have nonvanishing probability for small n_k (if $n_k = 1$, this probability is 16%). This is clearly wrong since in counting experiments, $\mu \geq 0$. Indeed, if $n_k \geq 1$, then even $\mu = 0$ is clearly ruled out. Note also that $n_k = 0$ does not necessarily imply that $\mu = 0$: even if $\mu = 1$, counts will be zero in $1/e \approx 37\%$ of cases. Another problem is that the range $n_k \pm \sigma_k$ does not correspond to the 68% confidence interval for true μ when n_k is small. These issues are important when fitting models to small count data (assuming that the available data are already binned). This idea is explored in a Bayesian context in §5.6.6.

4.9. Selection Effects and Luminosity Function Estimation

We have already discussed truncated and censored data sets in §4.2.7. We now consider these effects in more detail and introduce a nonparametric method for correcting the effects of the selection function on the inferred properties of the underlying pdf.

When the selection probability, or selection function $S(x)$, is known (often based on analysis of simulated data sets) and finite, we can use it to correct our estimate $f(x)$. The correction is trivial in the strictly one-dimensional case: the implied true distribution $h(x)$ is obtained from the observed $f(x)$ as

$$h(x) = \frac{f(x)}{S(x)}. \tag{4.82}$$

When additional observables are available, they might carry additional information about the behavior of the selection function, $S(x)$. One of the most important examples in astronomy is the case of flux-limited samples, as follows.

Assume that in addition to x, we also measure a quantity y, and that our selection function is such that $S(x) = 1$ for $0 \leq y \leq y_{\max}(x)$, and $S(x) = 0$ for $y > y_{\max}(x)$, with $x_{\min} \leq x \leq x_{\max}$. Here, the observable y may, or may not, be related to (correlated with) observable x, and the $y \geq 0$ assumption is

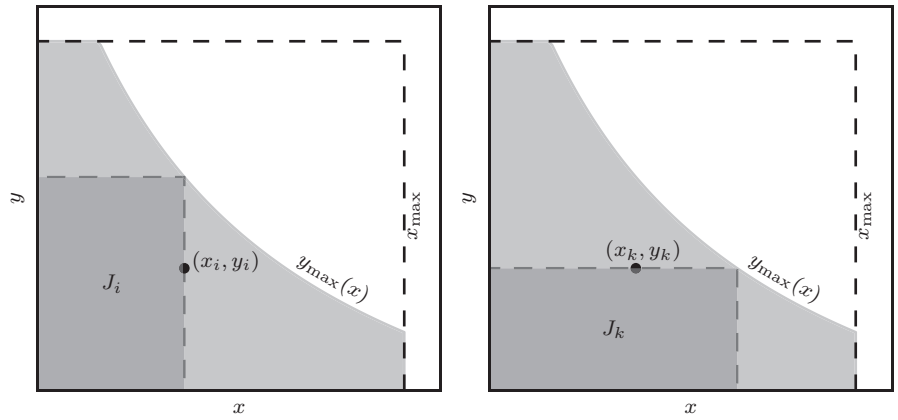

Figure 4.8. Illustration for the definition of a truncated data set, and for the comparable or associated subset used by the Lynden-Bell C^- method. The sample is limited by $x < x_{\max}$ and $y < y_{\max}(x)$ (light-shaded area). Associated sets J_i and J_k are shown by the dark-shaded area.

added for simplicity and without a loss of generality. In an astronomical context, x can be thought of as luminosity, L, (or absolute magnitude), and y as distance (or redshift in the cosmological context). The differential distribution of luminosity (probability density function) is called the luminosity function. In this example, and for noncosmological distances, we can compute $y_{\max}(x) = (x/(4\pi F_{\min}))^{1/2}$, where $F_{\min}$ is the smallest flux that our measuring apparatus can detect (or that we imposed on the sample during analysis); for illustration see figure 4.8. The observed distribution of x values is in general different from the distribution we would observe when $S(x) = 1$ for $y \leq (x_{\max}/(4\pi F_{\min}))^{1/2}$, that is, when the "missing" region, defined by $y_{\max}(x) < y \leq (x_{\max}/(4\pi F_{\min}))^{1/2} = y_{\max}(x_{\max})$, is not excluded. If the two-dimensional probability density is $n(x, y)$, then the latter is given by

$$h(x) = \int_0^{y_{\max}(x_{\max})} n(x, y)\, dy, \tag{4.83}$$

and the observed distribution corresponds to

$$f(x) = \int_0^{y_{\max}(x)} n(x, y)\, dy. \tag{4.84}$$

As is evident, the dependence of $n(x, y)$ on y directly affects the difference between $f(x)$ and $h(x)$. Therefore, in order to obtain an estimate of $h(x)$ based on measurements of $f(x)$ (the luminosity function in the example above), we need to estimate $n(x, y)$ first. Using the same example, $n(x, y)$ is the probability density function per unit luminosity *and* unit distance (or equivalently volume). Of course, there is no guarantee that the luminosity function is the same for near and far distances, that is, $n(x, y)$ need not be a separable function of x and y.

Let us formulate the problem as follows. Given a set of measured pairs (x_i, y_i), with $i = 1, \ldots, N$, and *known* relation $y_{\max}(x)$, estimate the two-dimensional distribution, $n(x, y)$, from which the sample was drawn. Assume that measurement

errors for both x and y are negligible compared to their observed ranges, that x is measured within a range defined by $x_{\min}$ and $x_{\max}$, and that the selection function is 1 for $0 \leq y \leq y_{\max}(x)$ and $x_{\min} \leq x \leq x_{\max}$, and 0 otherwise (for illustration, see figure 4.8).

In general, this problem can be solved by fitting some predefined (assumed) function to the data (i.e., determining a set of best-fit parameters), or in a nonparametric way. The former approach is typically implemented using maximum likelihood methods [4] as discussed in §4.2.2. An elegant nonparametric solution to this mathematical problem was developed by Lynden-Bell [18], and shown to be equivalent or better than other nonparametric methods by Petrosian [19]. In particular, Lynden-Bell's solution, dubbed the C^- method, is superior to the most famous nonparametric method, the $1/V_{\max}$ estimator of Schmidt [21]. Lynden-Bell's method belongs to methods known in statistical literature as the product-limit estimators (the most famous example is the Kaplan–Meier estimator for estimating the survival function; for example, the time until failure of a certain device).

4.9.1. Lynden-Bell's C^- Method

Lynden-Bell's C-minus method is implemented in the package `astroML.lumfunc`, using the functions `Cminus`, `binned_Cminus`, and `bootstrap_Cminus`. For data arrays `x` and `y`, with associated limits `xmax` and `ymax`, the call looks like this:

```
from astroML.lumfunc import Cminus
Nx, Ny, cuml_x, cuml_y = Cminus(x, y, xmax, ymax)
```

For details on the use of these functions, refer to the documentation and to the source code for figures 4.9 and 4.10.

Lynden-Bell's nonparametric C^- method can be applied to the above problem when the distributions along the two coordinates x and y are uncorrelated, that is, when we can assume that the bivariate distribution $n(x, y)$ is separable:

$$n(x, y) = \Psi(x)\, \rho(y). \tag{4.85}$$

Therefore, before using the C^- method we need to demonstrate that this assumption is valid.

Following Lynden-Bell, the basic steps for testing that the bivariate distribution $n(x, y)$ is separable are the following:

1. Define a *comparable* or *associated* set for each object i such that $J_i = \{j : x_j < x_i,\ y_j < y_{\max}(x_i)\}$; this is the largest x-limited and y-limited data subset for object i, with N_i elements (see the left panel of figure 4.8).
2. Sort the set J_i by y_j; this gives us the rank R_j for each object (ranging from 1 to N_i).
3. Define the rank R_i for object i in *its* associated set: this is essentially the number of objects with $y < y_i$ in set J_i.

4. Now, if x and y are truly independent, R_i must be distributed *uniformly* between 0 and N_i; in this case, it is trivial to determine the expectation value and variance for R_i: $E(R_i) = E_i = N_i/2$ and $V(R_i) = V_i = N_i^2/12$. We can define the statistic

$$\tau = \frac{\sum_i (R_i - E_i)}{\sqrt{\sum_i V_i}}. \tag{4.86}$$

   If $\tau < 1$, then x and y are uncorrelated at $\sim 1\sigma$ level (this step appears similar to Schmidt's $V/V_{\rm max}$ test discussed below; nevertheless, they are fundamentally different because $V/V_{\rm max}$ tests the hypothesis of a uniform distribution in the y direction, while the statistic τ tests the hypothesis of uncorrelated x and y).

Assuming that $\tau < 1$, it is straightforward to show, using relatively simple probability integral analysis (e.g., see the appendix in [10], as well as the original Lynden-Bell paper [18]), how to determine cumulative distribution functions. The cumulative distributions are defined as

$$\Phi(x) = \int_{-\infty}^{x} \Psi(x')\, dx' \tag{4.87}$$

and

$$\Sigma(y) = \int_{-\infty}^{y} \rho(y')\, dy'. \tag{4.88}$$

Then,

$$\Phi(x_i) = \Phi(x_1) \prod_{k=2}^{i} (1 + 1/N_k), \tag{4.89}$$

where it is assumed that x_i are sorted ($x_1 \le x_k \le x_N$). Analogously, if M_k is the number of objects in a set defined by $J_k = \{j : y_j < y_k, y_{\rm max}(x_j) > y_k\}$ (see the right panel of figure 4.8), then

$$\Sigma(y_j) = \Sigma(y_1) \prod_{k=2}^{j} (1 + 1/M_k). \tag{4.90}$$

Note that both $\Phi(x_j)$ and $\Sigma(y_j)$ are defined on nonuniform grids with N values, corresponding to the N measured values. Essentially, the C^- method assumes a piecewise constant model for $\Phi(x)$ and $\Sigma(y)$ between data points (equivalently, differential distributions are modeled as Dirac δ functions at the position of each data point). As shown by Petrosian, $\Phi(x)$ and $\Sigma(y)$ represent an optimal data summary [19].

The differential distributions $\Psi(x)$ and $\rho(y)$ can be obtained by binning cumulative distributions in the relevant axis; the statistical noise (errors) for both quantities can be estimated as described in §4.8.2, or using bootstrap (§4.5).

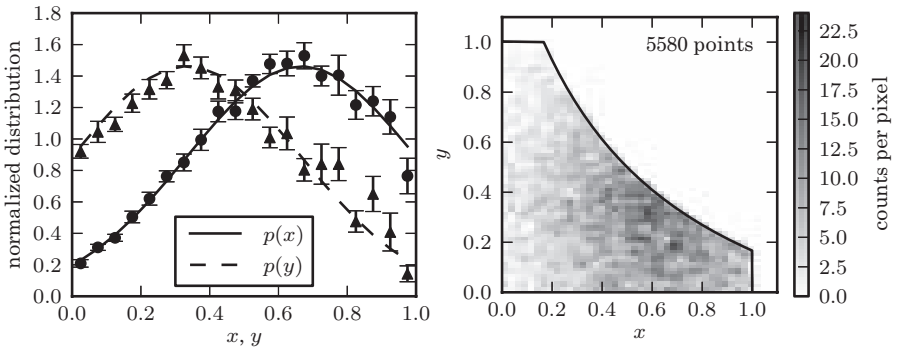

Figure 4.9. An example of using Lynden-Bell's C^- method to estimate a bivariate distribution from a truncated sample. The lines in the left panel show the true one-dimensional distributions of x and y (truncated Gaussian distributions). The two-dimensional distribution is assumed to be separable; see eq. 4.85. A realization of the distribution is shown in the right panel, with a truncation given by the solid line. The points in the left panel are computed from the truncated data set using the C^- method, with error bars from 20 bootstrap resamples.

An approximate normalization can be obtained by requiring that the total predicted number of objects is equal to their observed number.

We first illustrate the C^- method using a toy model where the answer is known; see figure 4.9. The input distributions are recovered to within uncertainties estimated using bootstrap resampling. A realistic example is based on two samples of galaxies with SDSS spectra (see §1.5.5). A flux-limited sample of galaxies with an r-band magnitude cut of $r < 17.7$ is selected from the redshift range $0.08 < z < 0.12$, and separated into blue and red subsamples using the color boundary $u-r = 2.22$. These color-selected subsamples closely correspond to spiral and elliptical galaxies and are expected to have different luminosity distributions [24]. Absolute magnitudes were computed from the distance modulus based on the spectroscopic redshift, assuming WMAP cosmology (see the source code of figure 4.10 for details). For simplicity, we ignore K corrections, whose effects should be very small for this redshift range (for a more rigorous treatment, see [3]). As expected, the difference in luminosity functions is easily discernible in figure 4.10. Due to the large sample size, statistical uncertainties are very small. True uncertainties are dominated by systematic errors because we did not take evolutionary and K corrections into account; we assumed that the bivariate distribution is separable, and we assumed that the selection function is unity. For a more detailed analysis and discussion of the luminosity function of SDSS galaxies, see [4].

It is instructive to compare the results of the C^- method with the results obtained using the $1/V_{\rm max}$ method [21]. The latter assumes that the observed sources are uniformly distributed in probed volume, and multiplies the counts in each x bin j by a correction factor that takes into account the fraction of volume accessible to each measured source. With x corresponding to distance, and assuming that volume scales as the cube of distance (this assumption is not correct at cosmological distances),

$$S_j = \sum_i \left(\frac{x_i}{x_{\rm max}(j)} \right)^3 , \tag{4.91}$$

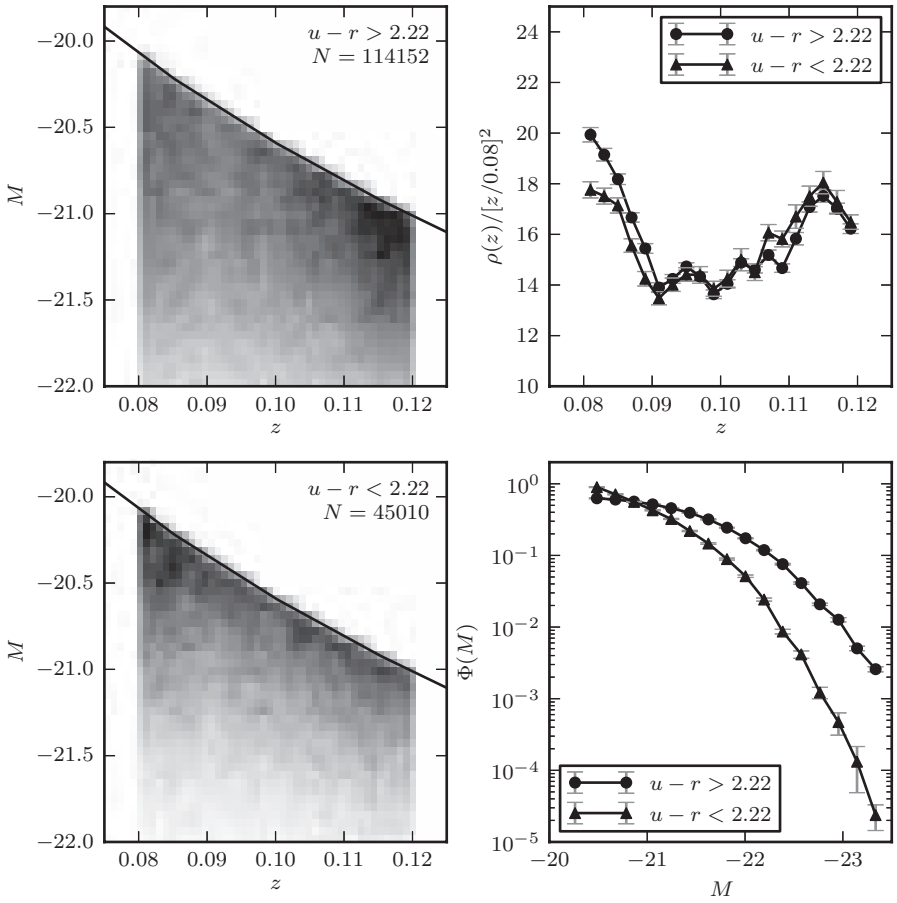

Figure 4.10. An example of computing the luminosity function for two $u - r$ color-selected subsamples of SDSS galaxies using Lynden-Bell's C^- method. The galaxies are selected from the SDSS spectroscopic sample, with redshift in the range $0.08 < z < 0.12$ and flux limited to $r < 17.7$. The left panels show the distribution of sources as a function of redshift and absolute magnitude. The distribution $p(z, M) = \rho(z)\Phi(m)$ is obtained using Lynden-Bell's method, with errors determined by 20 bootstrap resamples. The results are shown in the right panels. For the redshift distribution, we multiply the result by z^2 for clarity. Note that the most luminous galaxies belong to the photometrically red subsample, as discernible in the bottom-right panel.

where the sum is over all x_i measurements from y (luminosity) bin j, and the maximum distance $x_{\max}(j)$ is defined by $y_j = y_{\max}[x_{\max}(j)]$. Given S_j, h_j is determined from f_j using eq. 4.82. Effectively, each measurement contributes more than a single count, proportionally to $1/x_i^3$. This correction procedure is correct only if there is no variation of the underlying distribution with distance. Lynden-Bell's C^- method is more versatile because it can treat cases when the underlying distribution varies with distance (as long as this variation does not depend on the other coordinate).

Complicated selection function

In practical problems, the selection function is often more complicated than given by the sharp boundary at $y_{\max}(x)$. A generalization of the C^- method to the case of an arbitrary selection function, $S(x, y)$, is described in [10]. First define a generalized comparable set $J_i = \{j : x_j > x_i\}$, and then generalize N_i to the quantity

$$T_i = \sum_{j=1}^{N_i} \frac{S(x_i, y_j)}{S(x_j, y_j)}, \tag{4.92}$$

with a redefined rank

$$R_i = \sum_{j=1}^{N_i} \frac{S(x_i, y_j)}{S(x_j, y_j)}, \tag{4.93}$$

for $y_j < y_i$. It follows that $E(R_i) = T_i/2$ and $V(R_i) = T_i^2/12$, as in the case of a simple selection function.

4.9.2. A Flexible Method of Estimating Luminosity Functions

What can we do when measurement errors for x and y are not negligible, or when the bivariate distribution we want to infer is not a separable function of x and y? A powerful and completely general Bayesian approach is described by Kelly, Fan, and Vestergaard in [14]. They model the function $n(x, y)$ as a mixture of Gaussian functions (see §4.4). Although this approach is formally parametric, it is essentially as flexible as nonparametric methods.

4.10. Summary

In this chapter we have reviewed classical or frequentist techniques used for data modeling, estimating confidence intervals, and hypothesis testing. In the following chapter, we will build upon this toolkit by considering Bayesian methods. A combination of ideas from these two chapters will form the basis of the machine learning and data mining techniques presented in part III of the book.

References

[1] Archambeau, C., J. Lee, and M. Verleysen (2003). On convergence problems of the EM algorithm for finite Gaussian mixtures. In *European Symposium on Artificial Neural Networks*, pp. 99–106.

[2] Benjamini, Y. and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. 57*, 289.

[3] Blanton, M. R., J. Brinkmann, I. Csabai, and others (2003). Estimating fixed-frame galaxy magnitudes in the Sloan Digital Sky Survey. *AJ 125*, 2348–2360.

[4] Blanton, M. R., D. W. Hogg, N. A. Bahcall, and others (2003). The galaxy luminosity function and luminosity density at redshift z = 0.1. *ApJ 592*, 819–838.

[5] Bovy, J., D. W. Hogg, and S. T. Roweis (2011). Extreme deconvolution: Inferring complete distribution functions from noisy, heterogeneous and incomplete observations. *Annals of Applied Statistics 5*, 1657–1677.

[6] Davison, A. C., D. V. Hinkley, and G. Young (2003). Recent developments in bootstrap methodology. *Statistical Science 18*, 141–57.

[7] Dempster, A. P., N. M. Laird, and D. Rubin (1977). Maximum-likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B 39*, 1–38.

[8] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Ann. Statistics 7*, 1–26.

[9] Efron, B. and D. V. Hinkley (1978). Assessing the accuracy of the maximum likelihood estimator: Observed versus expected Fisher information. *Biometrika 65*, 457–87.

[10] Fan, X., M. A. Strauss, D. P. Schneider, and others (2001). High-redshift quasars found in Sloan Digital Sky Survey commissioning data. IV. Luminosity function from the Fall Equatorial Stripe Sample. *AJ 121*, 54–65.

[11] Franklin, J. (1972). *Biometrica Tables for Statisticians*. Cambridge University Press.

[12] Freedman, D. and P. Diaconis (1981). On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete 57*, 453–476.

[13] Hastie, T., R. Tibshirani, and J. Friedman (1993). *An Introduction to the Boostrap*. Chapman and Hall/CRC.

[14] Kelly, B. C., X. Fan, and M. Vestergaard (2008). A flexible method of estimating luminosity functions. *ApJ 682*, 874–895.

[15] Liddle, A. R. (2007). Information criteria for astrophysical model selection. *MNRAS 377*, L74–L78.

[16] Lilliefors, H. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association 62*, 399–402.

[17] Loredo, T. J. (2012). Bayesian astrostatistics: A backward look to the future. *ArXiv:astro-ph/1208.3036*.

[18] Lynden-Bell, D. (1971). A method of allowing for known observational selection in small samples applied to 3CR quasars. *MNRAS 155*, 95.

[19] Petrosian, V. (1992). Luminosity function of flux-limited samples. In E. D. Feigelson and G. J. Babu (Eds.), *Statistical Challenges in Modern Astronomy*, pp. 173–200.

[20] Roche, A. (2011). EM algorithm and variants: An informal tutorial. *ArXiv:statistics/1105.1476*.

[21] Schmidt, M. (1968). Space distribution and luminosity functions of quasi-stellar radio sources. *ApJ 151*, 393.

[22] Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika 66*, 605–610.

[23] Shapiro, S. S. and M. B. Wilk (1965). An analysis of variance test for normality (complete samples). *Biometrika 52*, 591–61.

[24] Strateva, I., Ž. Ivezić, G. R. Knapp, and others (2001). Color separation of galaxy types in the Sloan Digital Sky Survey imaging data. *AJ 122*, 1861–1874.

[25] Wu, C. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics 11*, 94–103.